

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Gestion de l'historique du dialogue dans un système de communication orale homme-machine finalisée en langage naturel

Druart, Pascal

*Award date:*  
1991

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Institut d'Informatique  
Rue Grandgagnage 21  
5000 Namur

**Gestion de l'historique du  
dialogue dans un système  
de communication orale  
homme-machine finalisée  
en langage naturel**

**Pascal DRUART**

Promoteur : J. BERLEUR s.j.

Mémoire présenté pour l'obtention  
du grade de licencié et maître en informatique

Année académique 1990-1991

# Table des matières

Résumé.....	1
Remerciements.....	2
Introduction.....	3
 <b>Chapitre 1 La communication orale homme-machine finalisée en langage naturel.....</b>	 <b>5</b>
1 La communication orale.....	5
2 La communication homme-machine.....	7
La situation de la communication.....	8
Le domaine de l'interaction ou l'univers.....	8
La population concernée.....	9
Le mode de communication.....	10
3 La communication finalisée en langage naturel.....	11
4 Le traitement automatique de la parole.....	12
La production de la parole.....	13
La perception de la parole.....	13
5 Les sources de connaissances.....	14
 <b>Chapitre 2 Présentation du système DIAL.....</b>	 <b>18</b>
1 Les objectifs du système DIAL.....	18
2 Le choix d'une application-test.....	19
3 L'architecture du système DIAL.....	20
3.1 Les composantes.....	21
3.2 Le flux d'informations.....	26
4 La composante DIALOG.....	28
4.1 L'interpréteur.....	29
4.2 Le raisonneur.....	30
4.3 Le dialogueur.....	32
5 La comparaison avec d'autres systèmes.....	34
5.1 Le système KEAL.....	34
5.2 Le système MYRTILLE I.....	34
5.3 Les systèmes HWIM et MYRTILLE II.....	35
5.4 Le système HEARSAY II.....	35

<b>Chapitre 3</b>	<b>La structure et l'historique du dialogue.....</b>	<b>3 7</b>
1	La structure du dialogue.....	3 7
1.1	Les problèmes liés au langage naturel.....	3 8
1.2	La représentation du dialogue.....	3 9
2	L'historique du dialogue.....	4 6
2.1	Le modèle de représentation des données.....	4 6
2.2	La gestion de l'historique de dialogue.....	6 7
<b>Chapitre 4</b>	<b>Les résultats et perspectives.....</b>	<b>7 0</b>
1	Les limitations .....	7 0
2	L'élaboration d'une application.....	7 0
2.1	Les explications.....	7 0
2.2	Des exemples.....	7 2
3	Les perspectives .....	7 6
	Conclusions.....	7 9
	Bibliographie.....	8 1
	Annexes.....	8 5

# Résumé

Dans un système de communication orale homme-machine finalisée en langage naturel, l'historique du dialogue tient une place importante car il permet au système de résoudre les diverses références du dialogue entre l'homme et la machine. Les données qui sont stockées dans cet historique doivent avoir une structure souple, efficace et adéquate pour permettre un traitement pragmatique de la part du système. Pour obtenir une structure de qualité, l'unité de référence, dans l'historique du dialogue, ne sera pas le concept, mais l'ensemble de concepts comportant des attributs nécessaires pour un traitement pragmatique. Ce dernier sera facilité par la présence d'une librairie de primitives permettant la recherche et la modification dans cette "base de données".

Le but de ce travail est d'asseoir les bases de cette structure de données et d'élaborer la librairie de fonctions qui y est associée.

# Abstract

In a task-oriented man-machine dialogue system, the dialogue history takes an important place, because it enables the system to resolve different references made during the dialogue between man and machine. The data which are stored in this history must have a flexible, efficient and adapted structure to allow the system's pragmatic processing. To obtain a structure with such qualities, the reference unit, in the dialogue history, will not be the concept, but the set of concepts with different attributes which are required for a pragmatic treatment. This treatment will be facilitated by the presence of a primitive's library that allows the search and the modification in this "data base".

The aim of this work is to lay the foundations of this data structure and to implement the associated functions library.

# Remerciements

Je tiens à exprimer mes remerciements les plus sincères :

à Jacques Berleur, Recteur des Facultés Universitaires Notre-Dame de la Paix de Namur qui, lors de son cours de "Méta-Informatique", a suscité mon intérêt pour ce domaine de l'informatique et qui, grâce à ses remarques et sa compétence, a permis la réalisation de ce travail.

à Jean-Marie Pierrel, Professeur de l'Université de Nancy I, qui m'a accueilli dans son équipe et a permis le bon déroulement de mon stage de troisième licence au Centre de Recherche en Informatique de Nancy.

à Azim Roussanaly, Maître de conférence et Chercheur au C.R.I.N., qui, grâce à ses explications et sa compétence dans ce domaine, a dirigé mes travaux et a permis la rédaction de ce manuscrit. Sans lui, rien n'aurait été possible.

à Guy Deville, Premier Assistant à l'Ecole des Langues Vivantes aux F.U.N.D.P. de Namur, pour son aide et ses remarques pertinentes dans le vaste domaine de la linguistique.

à tous les membres de l'équipe PAROLE-DIALOGUE qui m'ont accueilli et aidé lors de mon stage de fin d'études au C.R.I.N.

à ma famille et à tous mes amis qui m'ont aidé et soutenu lors de mes études.

# Introduction

Le domaine de la communication orale homme-machine est un domaine très vaste dans lequel travaillent beaucoup de personnes. Elles se sont rendues compte d'un grand nombre de difficultés lors de leurs recherches, que ce soit au point de vue linguistique ou informatique. C'est pourquoi certaines d'entre elles ont décidé de créer une équipe pluridisciplinaire de linguistes et d'informaticiens afin de valider les solutions pratiques avec les différents concepts théoriques d'un modèle de dialogue.

Différents projets ont vu le jour dans divers laboratoires d'informatique. Ainsi naquirent au Centre de Recherche en Informatique de Nancy (CRIN) l'équipe DIALOGUE et le projet DIAL. Nos recherches sont axées sur la composante pragmatique du système DIAL et plus précisément sur la gestion et la représentation de l'historique du dialogue dans un tel système.

Nous avons pour cela adopté une méthodologie de travail rigoureuse indispensable pour affronter le domaine complexe qu'est la communication orale homme-machine. Cette méthode de travail est constituée de trois étapes : premièrement, une approche théorique à partir de diverses lectures sur différents modèles théoriques de dialogue. Deuxièmement, une approche empirique au cours de laquelle les hypothèses théoriques sont vérifiées sur base de divers corpus. Et troisièmement, une approche pratique où une application regroupant les différents concepts du modèle théorique retenus dans les deux étapes précédentes est mise en oeuvre.

Le projet DIAL n'est pas né en un jour. D'autres systèmes en ont été les précurseurs. Ils ont, d'ailleurs, en tentant de résoudre le problème de la communication orale homme-machine finalisée en langage naturel, dégagé leurs limites ainsi que mis à jour d'autres questions fondamentales sur un sujet en plein développement. Le niveau sémantique dans lequel était analysé le dialogue doit être dépassé pour faire place à une dimension pragmatique des énoncés de dialogue oraux entre un homme et une machine.

Le chapitre 1 situera le cadre général de nos recherches tandis que le chapitre 2 précisera le système dans lequel nous développerons la gestion et la représentation de l'historique du dialogue (chapitre 3). Cet historique est composé d'une structure de données et d'une librairie de primitives qui sont utilisées par les différents modules de traitement de la composante DIALOG pour réaliser un traitement pragmatique adéquat. Le chapitre 4 constitue nos résultats et perspectives à propos des recherches que nous avons effectuées pour ce travail.



# Chapitre 1

## La communication orale homme-machine finalisée en langage naturel

Le cadre de développement de ce travail nous amène à situer ce que nous entendons par la communication orale homme-machine finalisée en langage naturel.

### 1 La communication orale

La communication telle que nous l'envisageons repose sur le modèle de SHANNON illustré à la figure 1.1.

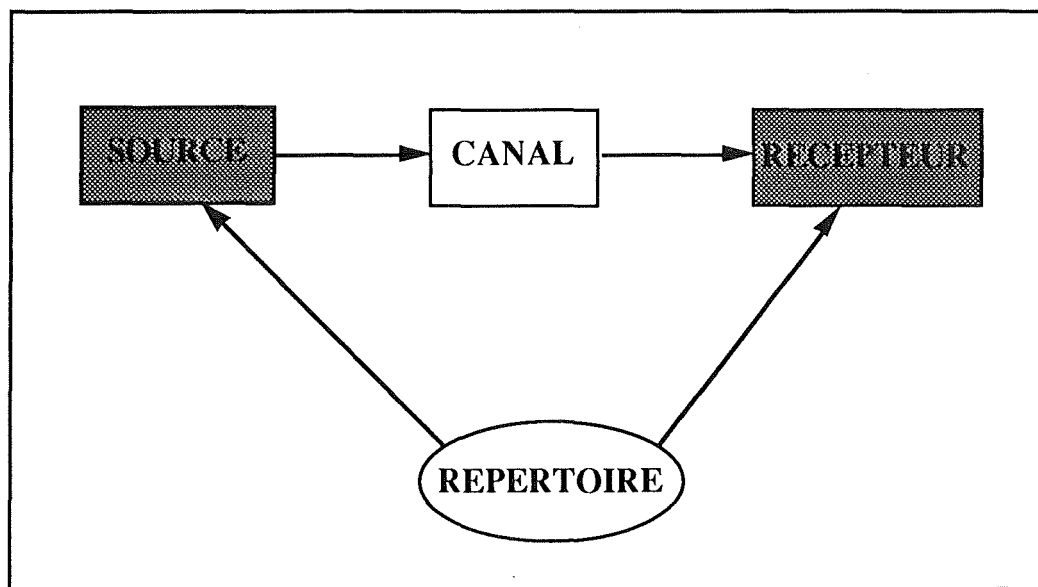


Figure 1.1 : *Modèle de la communication selon SHANNON*

Ce modèle définit la communication comme la transmission d'un message entre deux entités via un canal. Ces deux entités sont appelées source et récepteur.

Ces deux intervenants ont un répertoire commun, c'est-à-dire une ensemble de signes ainsi que leurs règles de combinaisons. Dans ce répertoire, la source sélectionne le message qu'elle désire envoyer. Ce dernier subit ensuite une opération d'encodage qui le rend apte à être transmis. Lors de la réception, le message est décodé et converti dans une forme qui peut alors être confrontée au répertoire du récepteur (voir figure 1.2) [HORMANN 72].

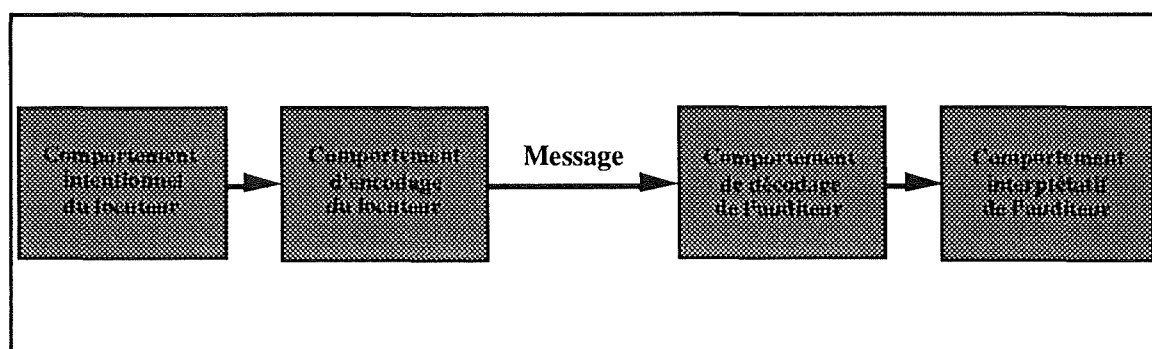


Figure 1.2 : *Modèle de la communication selon CARROLL*

Il s'agit cependant de modèles très généraux, car les messages peuvent être de types très variés (signes, images, ...).

Pour notre part, nous nous intéressons à la communication orale, c'est-à-dire que le message se présente sous forme de parole. Dès lors, des facultés très diverses interviennent, facultés variant de la capacité d'entendre et de parler à la capacité de comprendre, d'interpréter, de raisonner, de répondre et de mener un dialogue. Par ailleurs, comme nous nous situons dans le cadre de la communication orale, nous pouvons préciser les termes source et récepteur comme respectivement locuteur et auditeur.

Nous reviendrons sur ces différents points lors de la présentation du système DIAL (chapitre 2) et de la structure du dialogue (chapitre 3).

## 2 La communication homme-machine

Le système dans lequel ce travail s'est développé est un système de communication homme-machine, c'est-à-dire un système dans lequel un des intervenants (source ou récepteur) est un être humain, l'autre étant une machine [MOUSEL 90].

Le fait que l'un des deux interlocuteurs soit une machine a une répercussion sur le niveau du langage. Trois niveaux sont communément distingués :

**LF (Langages Formels)** : il s'agit des langages de programmation ou de commandes utilisés par des spécialistes à des fins bien précises. Ces langages (Ex : Cobol, Pascal, C, ...) sont traités par les techniques de compilation habituelles.

**LACN (Langages Artificiels à Consonance Naturelle)** : ce sont des langages dont la syntaxe est formellement définie et dont le traitement est comparable à celui des LF. La consonance naturelle intervient à différents niveaux :

- dans le lexique composé de mots de la langue naturelle.
- dans la restriction des structures des messages et des commandes acceptables à un sous-ensemble de la syntaxe du langage naturel. Selon la taille de ce dernier, le langage utilisé est syntaxiquement très contraint (sous-ensemble peu étendu) ou, à l'opposé, quasi naturel (sous-ensemble très étendu).

**SL (Sous-Langage)** : Ce sont des langages où les contraintes sont d'ordre sémantico-pragmatique. La notion de sous-langage n'étant pas clairement définie -différents points de vue se retrouvent dans la littérature-, nous adopterons celle de Deville-Paulussen qui définissent le sous-langage comme

un ensemble d'énoncés liés par un sujet limité, utilisés pour une fonction particulière et engendrés par une grammaire et un vocabulaire spécifique [DEVILLE 86].

Dans le cadre de notre travail, nous avons cependant limité le champ de la communication aux interfaces entre un utilisateur et une tâche. Nous ne nous intéresserons donc pas aux spécificités de la communication lorsqu'il s'agit d'un débat ou d'un interview (cfr les actes de dialogue dans le chapitre 3).

Les spécificités de la communication homme-machine ayant des influences sur le langage ont des origines variées telles que :

### **La situation de la communication**

L'utilisateur se place dans une situation de requête ou de commande, ce qui implique :

- sur le plan syntaxique, la réduction de la variété des structures des messages.
- sur le plan sémantique, la restriction des ambiguïtés dues aux ellipses et aux anaphores.
- sur le plan du dialogue, la présence d'une structure du dialogue liée à celle de la tâche (ou dialogue opératif).

### **Le domaine de l'interaction ou l'univers**

L'utilisateur, lorsqu'il formule sa requête ou sa commande se situe dans un univers restreint délimité par l'application qu'il a choisie. Cette restriction a une influence sur le vocabulaire et également sur :

**la taille du lexique** : il existe une relation entre la taille du vocabulaire et l'application. Prenons deux exemples : le premier, une application concernant les renseignements administratifs (formalités pour les pièces administratives), le

nombre de mots utilisés est d'environ 1000 [ROUSSANALY 86]. Le second, une application de renseignements administratifs (Ministère de l'emploi et du travail) [DEVILLE 87] où la taille du lexique est de plus ou moins 1400 mots.

**La fréquence des mots utilisés :** 90 % des mots utilisés lors d'une conversation entre les pilotes et les contrôleurs aériens représente près de 25 % du lexique [FALZON 84], et ce en raison du jargon professionnel utilisé.

**La réduction des polysémies :** un même mot peut avoir plusieurs sens différents. Mais, lorsqu'on se situe dans le cadre d'une application bien précise, le mot utilisé renvoie dans la grande majorité des cas à la même signification.

### **La population concernée**

Lors de la communication naturelle avec un usager, la production verbale augmente si celui-ci possède peu de connaissances dans le domaine de sa requête. Un effort d'explications supplémentaires est donc nécessaire pour que l'interlocuteur novice obtienne satisfaction.

Dans le cadre de la communication homme-machine que nous envisageons ici, les utilisateurs peuvent être regroupés en trois classes selon leur niveau d'expertise :

**Les spécialistes :** ce groupe nous intéresse moins, car les applications élaborées pour leurs besoins sont plus du ressort du génie logiciel.

**Les professionnels ou les locuteurs entraînés :** ils communiquent de façon relativement fréquente avec l'ordinateur dans un langage préalablement appris.

**Les utilisateurs naïfs ou le grand public :** il s'agit d'utilisateurs occasionnels voulant réaliser une tâche précise et utilisant pour cela le langage naturel.

Ces trois groupes d'utilisateurs correspondent plus ou moins aux trois niveaux de langages mentionnés auparavant (voir figure 1.3) [ROUSSANALY 88].

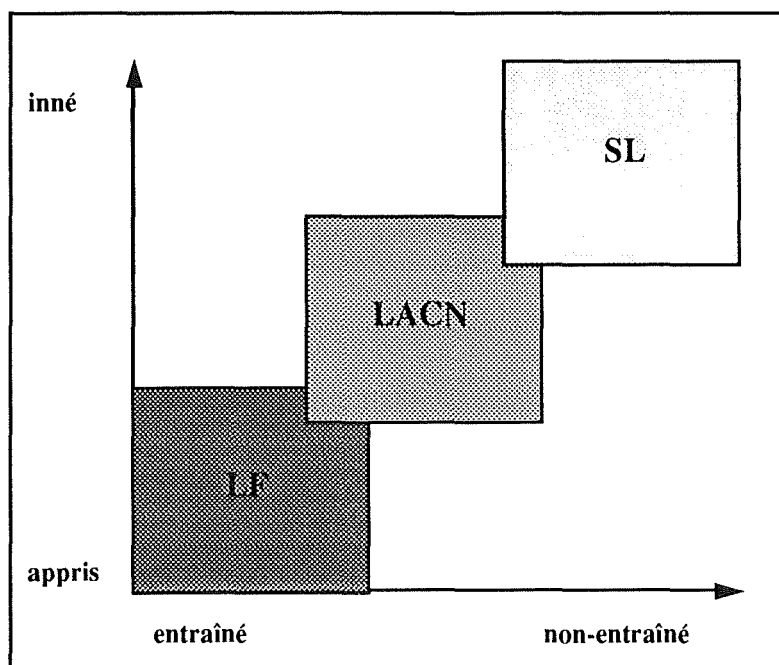


Figure 1.3 : *Relation entre la population et le langage*

## Le mode de communication

La présence d'une machine constitue un facteur psychologique qui influence le comportement langagier de l'utilisateur [GRECO 85]. D'après deux études menées en France, on constate :

- une restriction du vocabulaire utilisé.
- une augmentation du nombre d'interventions dans la situation d'une communication homme-machine.
- une complétude dans les demandes initiales de l'usager ainsi qu'une diminution de la variété des structures de phrases employées.

La plupart du temps, lorsqu'un utilisateur communique avec un ordinateur, il le fait à l'aide d'un écran et d'un clavier. Les messages transmis le sont alors sous forme textuelle et sont très fiables. La parole étant le moyen le plus courant de communication chez l'homme, il devient indispensable de disposer d'interfaces vocales pour la transmission des messages vers la machine, interfaces dont la fiabilité n'est encore que très limitée.

Notons également qu'il existe une différence entre le français écrit et le français parlé. Elle est accentuée par la présence lors de l'oral de phénomènes tels que des hésitations, des reprises et des messages de maintien du canal de communication (cfr actes de dialogue chapitre 3). Ces phénomènes n'existent pour ainsi dire pas à l'écrit. L'emploi de messages spécifiques tels que des demandes de répétition, de reformulation (en cas d'ambiguïté ou de défaillance du canal de communication) est accentué par le fait que la reconnaissance du signal acoustique n'est pas de 100 %. Actuellement, elle se situe aux alentours de 70 % [ROUSSANALY 88]. Nous pouvons donc parler ici d'une influence matérielle sur le langage.

### **3 La communication finalisée en langage naturel**

La communication, telle que nous la définissons dans ce travail, doit être la plus naturelle possible. C'est pourquoi une limitation a priori du répertoire est inadmissible. Mais envisager un système de communication en langage naturel est à l'heure actuelle totalement utopique. Nous pouvons tout au plus parler d'un système qui est le plus proche possible de la "langue naturelle".

Le fait d'avoir défini notre système de travail comme un système de communication finalisée (en langage naturel) va nous permettre de poser certaines limitations "raisonnables" . Il ne peut

être question de limiter la complexité du langage traité. Le traitement de la parole continue et l'indépendance du système sont autant d'obligations qui ne peuvent pas être remises en cause puisque nous voulons que le système mis en oeuvre soit accessible au grand public.

Les seules restrictions que nous nous sommes imposées se situent en fait au niveau des situations de communication. Notre intérêt s'est porté exclusivement sur les dialogues dont le but est la réalisation d'une tâche. Par ailleurs, on suppose de la part du système une faculté de compréhension. Nous y reviendrons plus tard.

C'est donc dans ce sens que nous entendons le concept de communication finalisée en langage naturel. Nous retiendrons d'ailleurs la définition de P. Mousel : la communication finalisée est celle dont le but est la réalisation d'une tâche [MOUSEL 90].

Après avoir explicité ce que nous entendons par communication orale homme-machine finalisée en langage naturel, il nous apparaît important d'introduire quelques notions relatives au traitement automatique de la parole, car de nouvelles contraintes d'acquisition et d'émission de message ont surgi du fait de l'utilisation d'une interface vocale.

## **4 Le traitement automatique de la parole**

Comme nous avons choisi de travailler dans le domaine de la communication orale, le système qui en découle doit posséder à la fois des facultés de production et de perception de la parole.



## **La production de la parole**

Nous trouvons actuellement sur le marché des synthétiseurs très performants qui peuvent émettre tous les sons du système phonatoire humain. De plus, ils peuvent générer des phrases entières avec une intelligibilité acceptable s'ils sont interfacés avec des logiciels adaptés.

Cependant, il ne faut pas croire que les problèmes de production de la parole en soient résolus pour autant. A cause de la subjectivité que peut contenir un message, il existe encore différents problèmes liés aux choix complexes à faire lors du passage de la représentation conceptuelle à la forme acoustique de ce message. Citons deux de ces problèmes qui nous semblent représentatifs de la complexité de ces choix :

**Le choix de la formulation** : selon la situation, des mots ou des structures de phrases sont plus appropriés que d'autres.

**La pertinence du message** : un message peut véhiculer plusieurs informations. Lesquelles doivent être explicitées ou laissées implicites ?

## **La perception de la parole**

Dans les années 60, la recherche en matière de perception s'est axée sur la reconnaissance des mots isolés. Son fonctionnement reposait sur un modèle de comparaison du mot à identifier avec un ensemble de formes de référence mémorisées par un apprentissage préalable.

Les années 70 et l'utilisation de contraintes linguistiques ont permis de franchir un pas supplémentaire dans ce domaine

(système pouvant saisir un programme en Algol prononcé mot à mot).

A l'heure actuelle, dans le domaine de la reconnaissance des mots isolés, les recherches s'orientent vers l'aspect multilocuteur (indépendance du système) et vers l'accroissement de la taille du vocabulaire.

Parallèlement, certains se sont attachés à réaliser des systèmes automatiques de compréhension de la parole continue. Commencé en 1971, le projet ARPA-SUR a permis de débloquent des fonds pour la recherche dans ce domaine. Plusieurs systèmes ont été développés par différents laboratoires dans le cadre de ce projet dont:

**SRI-SDC** par le Stanford Research Institute et la System Development Corporation

**HWIM** (Hear What I Mean) par la firme Bolt Beranek and Newman

**HEARSAY I et II** et **HARPY** par la Carnegie-Mellon University

Ces réalisations ont eu des effets catalyseurs aux USA, en Europe et au Japon. Actuellement, les systèmes développés évoluent vers la diversification des connaissances ainsi que vers une plus grande coopération entre celles-ci.

## **5 Les sources de connaissances**

La communication orale, comme nous l'avons dit, implique de la part du système diverses capacités. Pour mettre en oeuvre les processus de compréhension et d'interprétation, plusieurs niveaux

de connaissances doivent être mis en jeu. D'après Liberman, ils sont organisés hiérarchiquement (voir figure 1.4) [LIBERMAN 70]. Ces niveaux sont, dans l'ordre, l'acoustique, la phonétique, la phonologique, la prosodie, le lexique, la syntaxe, la sémantique et la pragmatique.

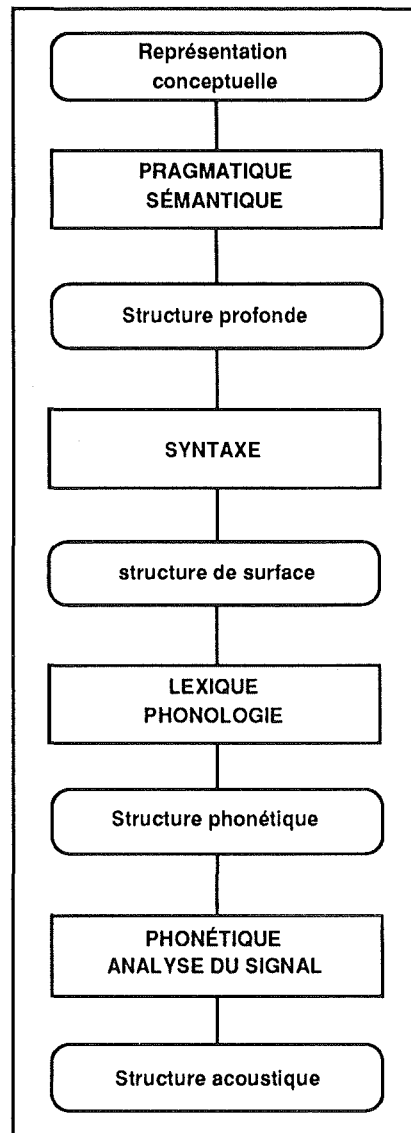


Figure 1.4 : *Un modèle de la perception de la parole d'après Liberman*

Pour notre part, dans le traitement automatique de la parole, nous nous intéresserons surtout à leur rôle :

**Le niveau acoustique** : il s'agit de connaissances permettant de détecter les débuts et fins de messages ; il

s'agit aussi d'extraire des paramètres du signal, informations qui seront traitées à des niveaux supérieurs.

**Le niveau phonétique** : ce sont des connaissances qui concernent les différentes unités phonétiques apparaissant dans la parole et ce, indépendamment du contexte d'émission.

**Le niveau phonologique** : ces connaissances complètent celles du niveau phonétique car elles prennent en compte le contexte. Elles rendent compte des spécificités individuelles (accent, sexe, ...) qui provoquent des altérations du signal et elles contiennent des phénomènes propres à la langue (coarticulation, liaison).

Ces trois premiers niveaux forment ce qu'on appelle le décodage acoustico-phonétique (DAP).

**Le niveau prosodique** : il s'agit de connaissances qui sont utilisées pour détecter les frontières de syntagmes (mots ou groupes de mots) grâce à des marqueurs prosodiques. Les processeurs lexicaux et syntaxiques exploiteront ces informations pour déterminer si la phrase prononcée est affirmative ou interrogative.

**Le niveau lexical** : il contient l'ensemble des mots de vocabulaire. Une transcription phonétique leur est respectivement associée ainsi qu'un nombre de traits syntaxiques et sémantiques. Lors de la compréhension, selon les hypothèses reçues des niveaux supérieurs, le processeur lexical sélectionne un sous-ensemble de mots du lexique et vérifie leur présence sur le treillis phonétique.

**Les niveaux syntaxique et sémantique** : ces connaissances constituent toutes les deux les règles régissant l'agencement des mots dans une phrase. La syntaxe détermine l'ordre des mots et les décrit à l'aide de catégories syntaxiques tandis que la sémantique porte sur la combinaison des mots et les caractérise en fonction de leurs propriétés sémantiques.

**Le niveau pragmatique** : il s'occupe de l'interprétation d'un message en fonction du contexte créé par la tâche et le

dialogue. Il comprend la résolution de diverses références, la prise en compte de présuppositions, la détermination d'informations implicites ainsi que l'enchaînement du dialogue.

Après avoir présenté les différents niveaux de connaissances nécessaires à la compréhension et à l'interprétation d'un message lors de la communication orale homme-machine finalisée en langage naturel, détaillons le système de référence sur lequel est basé notre travail.

## Chapitre 2

# Présentation du système DIAL

## 1 Les objectifs du système DIAL

L'objectif du projet DIAL est la construction d'un système informatique susceptible de constituer une interface orale homme-machine efficace et confortable pour l'utilisateur dans le cadre d'applications informatiques destinées au grand public [GRICE 75] [CARBONNEL 84]. Il s'agit donc d'une communication entre une machine et un utilisateur naïf qui veut réaliser une tâche. Le sujet adopte à cet effet, un comportement coopératif, c'est-à-dire, qu'il

- ne donne ni trop, ni trop peu d'informations,
- est sincère et parle à bon escient,
- est pertinent,
- est facilement compréhensible.

A part ce principe général de coopération, le locuteur n'est soumis à aucune forme de conditionnement préalable. Il ne doit donc pas effectuer d'apprentissage.

A partir d'un dialogue oral finalisé qu'il devra en partie au moins gérer et structurer et d'une expertise dans un domaine relativement simple et circonscrit, le système devra être capable de comprendre et de satisfaire les demandes des usagers. Ces demandes seront exprimées de façon libre et en français.

Différentes applications pourront alors être mises en oeuvre et pourraient avoir comme but, par exemple,

- la consultation de bases de données météorologiques, administratives, téléphoniques, etc.
- les demandes de rendez-vous, de réservation et de location.
- les commandes d'articles dans les catalogues de vente par correspondance.

## 2 Le choix d'une application-test

Afin de tester le système, la consultation des informations administratives contenues dans les pages roses de l'annuaire téléphonique des P&T a été retenue. La figure 2.1 illustre un exemple du type d'informations que l'on peut y trouver.

**Cartes et pièces d'identité**

**1. La carte nationale d'identité**

Si vous êtes de nationalité française, vous pouvez obtenir une carte nationale d'identité.

**Pour l'obtenir, adressez-vous** :

- . au commissariat de police de votre domicile
- . ou à défaut, à la mairie.

**Formalité** :

Vous présentez vous-mêmes les pièces suivantes :

- . le formulaire qui vous aura été remis, après l'avoir rempli
- . deux photographies d'identité de 3,5x4 cm vous représentant de face, la tête nue,
- . votre livret de famille ou fiche d'état civil, etc.

Figure 2.1 : Un extrait des pages roses de l'annuaire des PTT

Pourquoi ce choix ? Différentes raisons l'expliquent :

- la population concernée est le grand public
- les tâches à réaliser sont simples, variées et relativement bien structurées
- la taille du lexique (1000 mots) est réaliste pour un traitement en temps réel [ROUSSANALY 86]
- le dialogue est finalisé avec un contrôle partagé du dialogue
- la grande diversité est présumée dans les interventions vu le nombre élevé des échanges

Il faut cependant noter que la structure des énoncés peut varier fortement en fonction des locuteurs et que les reprises et les hésitations sont fréquentes, difficiles à détecter et à intégrer dans des structures avec les méthodes d'analyse classique.

Le système doit, entre autres, faire face aux exigences des spécificités de l'oral (cfr chapitre 1). Il doit être capable de gérer les contestations du locuteur, de demander des reformulations, des précisions, des confirmations, des répétitions et si nécessaire, des consignes en rapport avec le canal de communication.

### **3 L'architecture du système DIAL**

L'architecture du système a été structurée en fonction d'une approche de type multi-bases de connaissances. Cela se traduit par une organisation modulaire et une base de connaissances spécifiques à chaque module. Le système possède donc les principes de modularité (pour une mise au point plus facile et une réalisation en parallèle de chaque module) et de flexibilité (différents niveaux



de paramétrisation rendent la mise en place de l'application plus aisée). Un corpus a, par ailleurs, été réalisé pour valider le modèle de représentation des connaissances et pour élaborer les connaissances nécessaires à l'application.

### **3.1 Les composantes**

Le système est composé de cinq modules : APHON pour le décodage acoustico phonétique, PROSO pour la prosodie, LEX pour le traitement lexical, SYN-SEM pour la syntaxe et la sémantique et enfin DIAL pour le dialogue. Cette dernière composante sera plus détaillée dans la section suivante car elle constitue la base même de notre travail.

#### **A APHON**

Premier maillon du processus de compréhension, le module APHON construit la description phonétique d'un énoncé à partir du signal acoustique numérisé. Le résultat est un treillis de phonèmes représenté par des segments. Une liste de phonèmes les plus probablement prononcés par le locuteur est associée à chaque segment.

La réalisation de cette composante (approche système expert) repose sur la reconstitution du raisonnement d'experts en linguistique [FOHR 86]. A partir de la représentation visuelle d'un spectrogramme d'un énoncé, on établit sa description phonétique.

La composante est donc formée :

- d'un ensemble de modules de prétraitements qui effectuent une segmentation du signal
- d'une base de connaissances contenant l'ensemble des règles d'expertise
- d'un moteur d'inférence.

La version actuelle du système expert donne un taux moyen de reconnaissance de 70 %.

## **B PROSO**

PROSO détient un double rôle dans le système. D'une part, il contribue à la détermination d'une partie de la signification de l'énoncé par l'émission d'hypothèses concernant sa forme (interrogatif, énonciatif, exclamatif, impératif). D'autre part, il essaie de localiser les marqueurs prosodiques indiquant soit une frontière de mot (lexical), soit une frontière de syntagme (syntaxique).

Le module PROSO doit retenir uniquement les marqueurs non-ambigus, c'est-à-dire ceux qui sont détectés avec un taux de certitude suffisamment élevé [BONIN 87].

## **C LEX**

Son objectif est de détecter les mots dans un treillis de phonèmes. Il opère de deux façons. Soit, il effectue une vérification de mot, suite à une hypothèse émise par un processus de niveau supérieur. Cette hypothèse est un couple composé d'une forme de référence (transcription phonétique du mot-hypothèse) et d'un repère temporel indiquant le début du mot dans le treillis de phonèmes. Le traitement consiste en un appariement de la forme de référence et du sous-treillis correspondant au mot à vérifier. La composante LEX alors valide ou rejette l'hypothèse traitée. Soit, il n'y a pas d'information temporelle sur un mot à valider et donc un procédé de recherche itérative est déclenché par la mise en route d'un balayage du treillis de phonèmes. A chaque itération, un appariement entre le mot de référence et la partie sélectionnée dans le treillis est tenté. Le résultat n'est plus un booléen (validation ou rejet de l'hypothèse-mot), mais un score de reconnaissance, c'est-à-dire une mesure de certitude de l'apparition du mot de référence [MANGEOL 89].

## D SYN-SEM

Le but du module syntaxico-sémantique SYN-SEM est de fournir une ou plusieurs structures syntaxico-sémantiques correspondant à un énoncé en langage naturel. La représentation retenue pour ces structures trouve son origine dans le modèle des grammaires de cas. Le module SYN-SEM utilise un modèle linguistique de type casuel, inspiré de la grammaire fonctionnelle de Dik [DIK 79]. Ce modèle propose une représentation sémantique de chaque énoncé selon une structure composée d'un prédicat (dérivé d'une primitive) et d'un nombre d'arguments de cette primitive. Ses termes sont spécifiés au moyen de fonctions sémantiques (cas) définies comme l'expression d'une relation sémantique entre un terme et une primitive [DEVILLE 89]. La structure casuelle d'une primitive est donc une séquence de cas spécifiques (obligatoires ou optionnels). La figure 2.2 nous donne un résumé des primitives prédictives du sous-langage de DIAL avec leur schéma de cas associés.

PRIMITIVE	CTRL DYN ATTR SPAT COGN TRS DITRS	CASE FRAME	EXAMPLE VERBS
ACT_1	+ + - - - - -	Agent	pointer
ACT_2	+ + - - - + -	Agent Theme	éditer
ACT_3	+ + - - - - +	Agent Theme Referent	associer
ECHPROD_3	+ + + - - - +	Agent Theme Benefactive	donner acquiescer
MVMT_1	+ + - + - - -	Agent	se balader
MVMT_2	+ + - + - + -	Agent Source/Direction/path	aller venir
MVMT_3	+ + - + - - +	Agent Theme Source/Direction/Path	mettre
COGNT_1	+ + - - + - -	Agent	réfléchir
COGNT_2	+ + - - + + -	Agent Theme	cher regarder
COGNT_3	+ + - - + - +	Agent Theme Cognizant	dire montrer
STATUT_1	- - - - - - -	Theme	être mineur
STATUT_2	- - - - - + -	Theme Referent	resembler
POSSESS_2	- - + - - + -	Benefactive Theme	posséder
LOCATION_2	- - - + - + -	Theme Locative	se trouver
LOCATION_3	+ - - + - - +	Positioner Theme Locative	tenir
ST_COGNT_2	- - - - + + -	Cognizant Theme	savoir
PROCESS_1	- + - - - - -	Processed	fonctionner
PROCESS_2	- + - - - + -	Effector Theme	abîmer
ECHOBT_2	- + + - - + -	Benefactive Theme	recevoir
PR_MVMT_1	- + - + - - -	Processed	tomber
PR_MVMT_2	- + - + - + -	Processed Source/Direction/Path	aboutir
PR_MVMT_3	- + - + - - +	Effector Theme Source/Direction/Path	déplacer
PR_COGNT_2	- + - - + + -	Cognizant Theme	voir oublier

Figure 2.2 : Liste des primitives prédictives et de leurs cas associés

Prenons, par exemple, la phrase :

"Comment est-ce qu'on se procure un passeport ?"

Le résultat de l'analyse est la structure illustrée à la figure 2.3

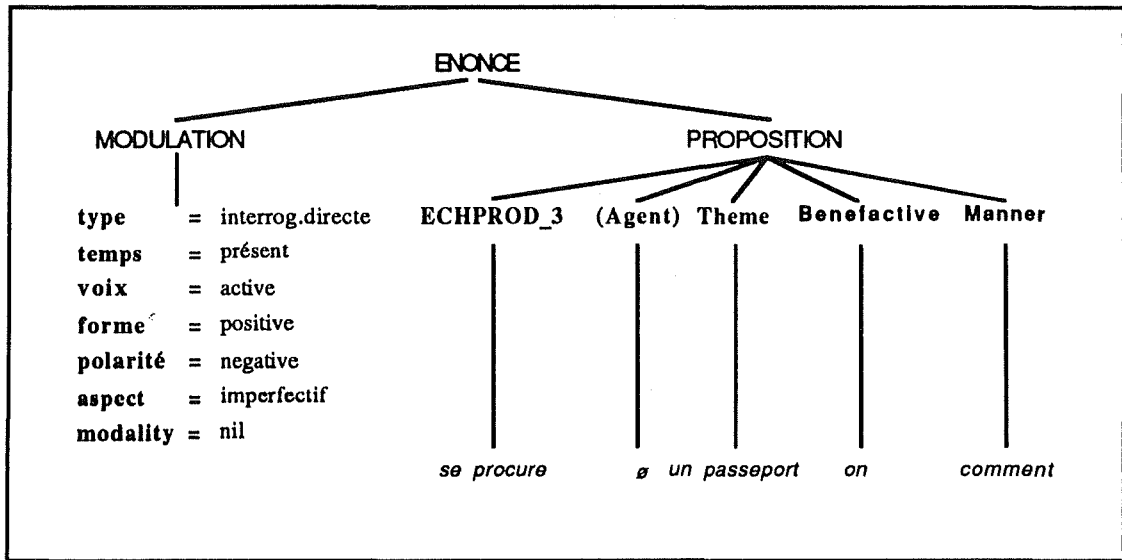


Figure 2.3 : Représentation sémantique de la phrase

" Comment est-ce qu'on se procure un passeport ? "

On pourra se reporter à [DEVILLE 89] pour détailler cette analyse.

Notons également que les traitements syntaxiques et sémantiques se déroulent en parallèle grâce à l'utilisation d'une base d'informations commune appelée espace des théories (pour plus de détails, voir [MOUSEL 90]).

## E DIALOG

L'objectif principal de la composante DIALOG est la structuration du dialogue. Ce module intègre pour cela différentes fonctionnalités. Tout d'abord, DIALOG effectue une interprétation contextuelle de l'énoncé en appliquant des règles d'interprétation contextuelle sur la représentation syntaxico-sémantique (venant de SYN-SEM) de l'énoncé ou sur les mots validés (venant de LEX). Cela permet d'homogénéiser les différentes représentations et de résoudre les références éventuelles du locuteur à des énoncés antérieurs.

Ensuite, une phase de raisonnement est déclenchée sur l'énoncé interprété en utilisant un modèle de la tâche. Une sélection des valeurs par défaut s'opère ainsi que la déduction des informations implicites et la gestion de l'état de la tâche (détermination des informations manquantes et détection des incohérences et des dépassements des limites du système).

La phase suivante est celle de gestion du dialogue. Elle permet à DIALOG de fournir une réponse au locuteur et d'émettre des prédictions sur sa prochaine intervention.

La communication entre ces différents traitements est effectuée par l'historique du dialogue. Mais nous y reviendrons dans la section suivante car la composante y sera plus détaillée.

Nous obtenons donc la hiérarchie générale du système (voir figure 2.4) [CARBONNEL 85]. Elle permet de concilier l'autonomie des processeurs, leur coopération et l'exploitation du parallélisme dans les traitements. Les différents flux d'informations qui existent entre les composantes du système méritent un mot d'explication.

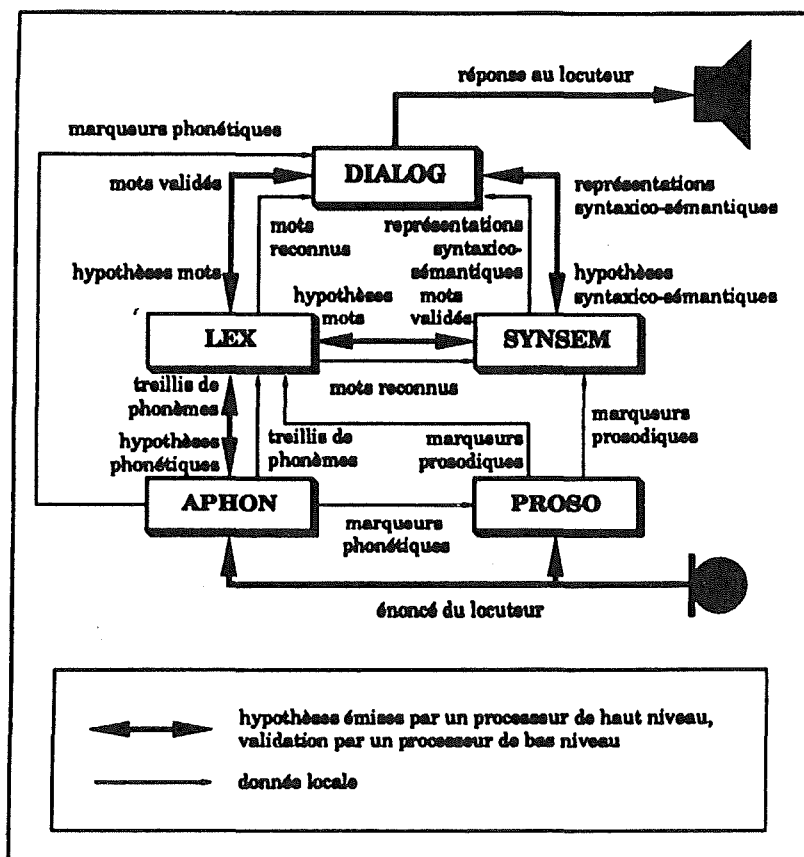


Figure 2.4 : *Hiérarchie générale du système DIAL*

### 3.2 Le flux d'informations

Chaque processeur (ou module) dispose de connaissances statiques propres. Les flèches reliant les différents modules représentent la circulation des informations à l'intérieur du système. Ces informations peuvent être de trois types :

**Les hypothèses émises par un processeur de niveau supérieur :** elles ont un caractère prédictif et demandent à être validées par un processeur de niveau inférieur. Il s'agit ici:

- des prédictions lexicales faites par DIALOG, c'est-à-dire lors de la sélection de mots-clefs dont

l'apparition dans l'énoncé suivant est fort probable.

- des prédictions syntaxico-sémantiques provenant du module DIALOG. Elles seront utilisées par SYN-SEM lors de son processus d'analyse.
- des hypothèses-mots provenant de SYN-SEM et validées par LEX.
- des structures phonétiques construites par LEX et correspondant à des mots. APHON vérifie leur présence directement sur le signal.

Toutes ces hypothèses induisent un fonctionnement descendant du système.

**Les représentations de l'énoncé :** un énoncé passe par plusieurs représentations qui sont propres à chaque processeur qui la traite (du signal acoustique à une représentation interprétative exploitable par DIALOG). APHON construit un treillis de phonèmes, LEX, lui, un treillis de mots, SYN-SEM un ensemble de structures syntaxico-sémantiques et DIAL une représentation interprétative de l'énoncé [ROUSSANALY 88].

**Les informations locales sur l'énoncé :** ce sont des données ponctuelles sur l'énoncé qui est traité. Il s'agit :

- des marques de segmentation : elles sont détectées par APHON et utilisées par PROSO pour la détection précise des marqueurs prosodiques.
- de la durée vocalique moyenne : elle est calculée par APHON pour ses traitements internes mais elle est également utilisée par PROSO pour le traitement prosodique du signal.

- des frontières de mots ou de syntagmes :  
détectées par PROSO, elles permettent à SYN-SEM d'éliminer des hypothèses syntaxiques et lexicales.
- des mots parfaitement reconnus : le processeur DIALOG les reçoit de LEX, ce qui lui permet de faire l'économie d'une analyse syntaxico-sémantique de SYN-SEM, les mots-clefs suffisant alors à la compréhension de l'énoncé.

## 4 La composante DIALOG

La composante DIALOG traite et gère le dialogue en cours. Elle le structure sur base des critères pragmatiques grâce à des échanges d'informations avec les autres composantes du système (voir le point 3.2 et la figure 2.5).

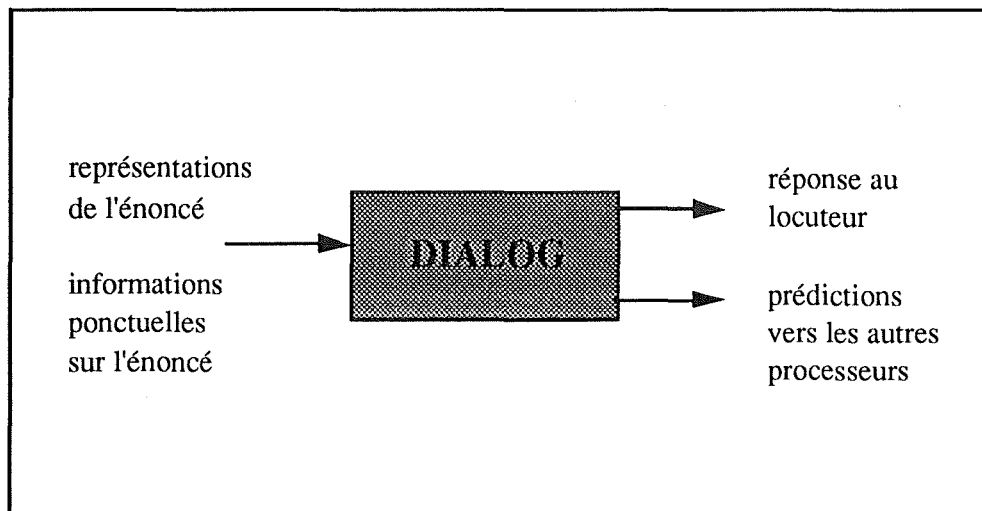


Figure 2.5 : *Structure externe de la composante DIALOG*

La composante pragmatique DIALOG est composée de trois modules de traitements : l'interpréteur, le raisonneur et le dialogueur (voir figure 2.6). Détaillons maintenant quels sont les rôles et les moyens de chacun de ces modules



## **4.1 L'interpréteur**

### **A Son rôle**

L'interpréteur fournit une interprétation contextuelle à partir d'un énoncé du locuteur, c'est-à-dire son but, ses faits et sa question. Il résout les références à l'historique pour avoir une représentation homogène de l'énoncé en cours d'interprétation.

En d'autres termes, il prend en charge toutes les informations destinées à DIALOG et les intègre à l'historique.

### **B Ses moyens**

L'interpréteur est un système conçu à base de règles de production. Il utilise une base de règles, une base de faits et une mémoire de travail ainsi qu'une interprétation.

- **La mémoire de travail** : elle dispose d'une représentation de toutes les informations en entrée. Chaque structure d'énoncé fournie par SYN-SEM est décomposée en prédicats du premier ordre stockés dans cette mémoire. Au cours de l'interprétation, l'activation de règles sélectionnées dans la base des règles (cfr infra) permet la déduction de nouveaux prédicats qui seront également stockés à cet endroit.
- **La base des règles** : ce sont des règles de format conditions-actions appelées règles d'interprétation. Les conditions sont appariées avec les prédicats de la mémoire de travail et avec les relations de la base des faits (cfr infra). Si cet appariement réussit, il entraîne le déclenchement d'actions telles que la suppression ou l'ajout d'un prédicat dans la mémoire de travail, la recherche d'une référence dans l'historique, ...etc.

- **La base des faits** : il s'agit de la mémoire à long terme de l'interpréteur. Elle a le même rôle que la mémoire de travail, excepté que son contenu reste invariant durant la période d'un dialogue. Elle regroupe des connaissances statiques d'un point de vue syntaxique, sémantique ou pragmatique. Ces connaissances décrivent les propriétés des éléments susceptibles d'apparaître en entrée.
- **L'interprétation** : structure initialement vide évoluant lors de l'interprétation. Elle contient l'interprétation contextuelle de l'énoncé à la fin du traitement interprétatif.

## 4.2 Le raisonneur

### A Son rôle

Le raisonneur propose l'ensemble des solutions susceptibles de satisfaire la requête de l'utilisateur. Cette opération met en oeuvre différents traitements :

- **La mise à jour de l'état de la tâche** : à partir de l'historique du dialogue et du modèle de la tâche, le raisonneur détermine, après chaque intervention du locuteur, l'état de la tâche (structure évolutive). Plusieurs situations peuvent se présenter : pas de solution, plusieurs solutions, la requête incomplète ou une seule solution satisfaisant la requête (solution idéale).
- **La gestion des informations implicites** : les conditions préalables à la satisfaction de la requête sont fournies de manière explicite ou implicite par les informations que donne l'utilisateur. Il existe deux types d'informations implicites : les faits déduits (par un raisonnement déductif) et les valeurs par défaut. Si une ou des valeurs par défaut sont utilisées pour la satisfaction de la requête de l'utilisateur, elles sont

mémorisées dans l'état de la tâche afin d'en informer le locuteur par la suite.

- **Le contrôle dynamique de la cohérence** : des informations contradictoires (dus à une reconnaissance incertaine) peuvent se glisser lors du dialogue entre la machine et le locuteur. Il faut donc envisager la détection de ces incohérences, leur résolution ou sinon leur mémorisation dans l'état de la tâche pour qu'elles soient transmises au dialogueur qui décide de la stratégie à adopter.

## **B Ses moyens**

Le raisonneur dispose d'un ensemble de structures d'informations et de leur module de traitements spécifiques. Il s'agit :

- **Des bases de connaissances statiques :**

- La base des règles : ensemble de règles régissant la déduction du raisonneur (pré-condition et post-condition).

- La base des renseignements : les renseignements sont composés d'un but (formule) et d'une procédure (condition alternative).

- La base des données : ce sont des formes dégénérées de renseignements (une formule et pas de condition alternative).

- **Des structures évolutives en fonction d'un dialogue :**

- La base des faits : structures incluses dans l'historique du dialogue. Un fait est représenté par une formule et des attributs tels que le score

de reconnaissance et l'état (défaut, donnée ou énoncé).

- La base des buts : structures incluses dans l'historique du dialogue dont le type d'actes de dialogue est un but (cfr actes de dialogue).

- La base des questions : structures incluses dans l'historique du dialogue dont le type d'actes de dialogue est une question (cfr actes de dialogue). Il en existe de deux types : question fermée ou question ouverte.

- **De la structure évolutive en fonction d'un énoncé :**

- L'état de la tâche : structure mémorisant les informations pertinentes données par le raisonneur et destinée au dialogueur. Ces informations sont organisées en rubriques telles que la question, le but, la réponse, l'incohérence, les valeurs par défaut.

## **4.3 Le dialogueur**

### **A Son rôle**

Le dialogueur décide de la réponse à apporter au locuteur et émet vers les autres composantes du système des hypothèses concernant l'énoncé suivant de l'utilisateur.

### **B Ses moyens**

Il possède des structures d'informations réparties autour d'un module de traitement. Il s'agit :

- **De l'état de la tâche (voir supra).**

- **De la représentation chronologique des énoncés :** suite d'interventions du locuteur et du système.

- **Des schémas de dialogue :** descriptions modulaires procédurales des situations typiques dans les dialogues à traiter par le système. Ces schémas sont la combinaison d'actions spécifiques et d'enchaînements à d'autres schémas. Par exemple, un schéma de dialogue peut être décomposé en trois phases : l'ouverture du dialogue, les négociations (autre schéma de dialogue) et clôture du dialogue.

La composante DIALOG est illustrée par la figure 2.6 [ROUSSANALY 88].

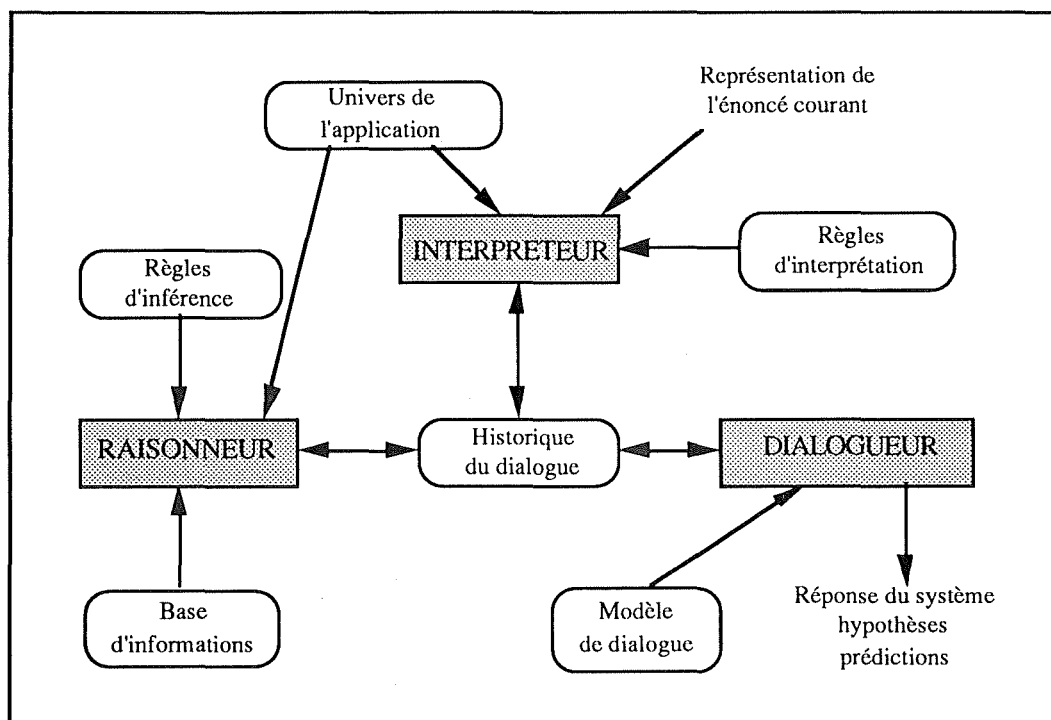


Figure 2.6 : Structure interne de la composante DIALOG

## **5 La comparaison avec d'autres systèmes**

Le projet DIAL n'est pas né en un jour. D'autres systèmes en ont été les précurseurs. Ils ont, d'ailleurs, en tentant de résoudre le problème de la communication orale homme-machine finalisée en langage naturel, dégagé leurs limites ainsi que mis à jour d'autres questions fondamentales sur un sujet en plein développement (voir chapitre 3 : les problèmes liés au langage naturel).

### **5.1 Le système KEAL**

Le CNET, à Lannion, a développé le système KEAL qui est un système de reconnaissance de parole continue. Une première version en fut implantée en 1979, avec pour objectif la reconnaissance de phrases. L'année 1981 donne lieu à une deuxième version du système qui est alors augmenté d'un module de gestion de dialogue. Ce module a pour but de permettre une communication homme-machine plus complète et naturelle [QUINTON 82].

Le système KEAL possède une architecture hiérarchique de type ascendant. Si les langages utilisés ont un vocabulaire de taille limitée, une telle organisation est bien adaptée. Cependant, si le vocabulaire employé est plus volumineux, le temps de traitement est pénalisé par la grande diversité du nombre d'hypothèses lexicales due à un manque d'interaction entre les différents modules de traitements internes.

### **5.2 Le système MYRTILLE I**

Dès le début des années 70, le C.R.I.N. a développé un système de compréhension automatique du discours continu appelé MYRTILLE I. L'objectif avoué était de construire un système général de reconnaissance du discours continu capable de traiter tous les langages définis en paramètre par une grammaire hors-contexte

[PIERREL 75]. Ce système était axé sur la reconnaissance de phrases. Il intégrait un module de dialogue qui posait dans le cas général de nombreux problèmes. C'est pourquoi la procédure de dialogue était considérée comme un traitement propre à chaque application, et donc elle devait être réécrite pour chaque nouvelle application.

MYRTILLE I possède une architecture hiérarchique de type descendant, ce qui convient plutôt bien à des systèmes traitant un langage artificiel à syntaxe très contrainte. Cependant, les traitements sémantiques et pragmatiques sont quelque peu éclipsés, traitements auxquels un intérêt particulier doit justement être apporté dans le cadre d'applications destinées au grand public.

### **5.3 Les systèmes HWIM et MYRTILLE II**

Ces deux systèmes sont une tentative de synthèse entre les deux approches ascendante et descendante qu'utilisent respectivement KEAL et MYRTILLE I. Ils possèdent tous les deux un module superviseur qui a pour but le séquençement de l'activation des autres modules. Ils intègrent également un mécanisme d'hypothèses-tests, ce qui augure de leur hiérarchie mixte [BOLT 76] [PIERREL 87].

Même si aucune architecture n'est imposée a priori aux différentes composantes des deux systèmes, le séquençement des tâches qui semblent indépendantes est cependant un inconvénient important.

### **5.4 Le système HEARSAY II**

Ce système, développé à la Carnegie-Mellon University, a tenté de remédier à l'inconvénient du séquençement des tâches (rencontré dans HWIM et MYRTILLE II) en considérant toutes ses composantes comme des modules indépendants. Chacun d'entre eux est appelé module de connaissances et communique avec les autres

via une structure centrale de données plus communément appelée "blackboard" (tableau noir). Elle contient toutes les hypothèses de tous les niveaux du système sur l'énoncé en cours de traitement. Les modules peuvent accéder à n'importe quelle hypothèse et sont respectivement activés par une modification dans le tableau noir [ERMAN 80].

L'organisation générale ainsi que l'indépendance des modules permet un développement incrémental de chacune des composantes du système. Mais l'architecture de HEARSAY II peut apparaître comme trop générale et, dès lors, peu efficace. En effet, il y a peu de chance qu'un jour on ajoute une composante supplémentaire au système, les sources de connaissances et leurs interactions étant bien maîtrisées aujourd'hui.



## Chapitre 3

# La structure et l'historique du dialogue

## 1 La structure du dialogue

Après avoir présenté la composante DIALOG et ses modules de traitements internes (interpréteur, raisonneur et dialogueur), il est important de voir comment un dialogue se structure afin de pouvoir envisager une structure de données-l'historique du dialogue-adaptée au stockage des informations recueillies lors de la communication orale entre une machine et un homme. Autour de cet historique, une librairie de fonctions d'accès et de constructeurs doit également être disponible. Grâce à cette librairie de fonctions, l'interpréteur, le raisonneur et le dialogueur pourront réaliser divers travaux qui leur incombent.

Mais, auparavant, il nous faut encore évoquer différents problèmes qui ont émergés lors de différentes expériences précédant le projet DIAL. Le fait d'envisager une communication orale finalisée en langage naturel soulève différentes questions auxquelles les modules de traitements internes de DIALOG pourront apporter une réponse grâce à l'utilisation d'une structure du dialogue adéquate et précise.

## **1.1 Les problèmes liés au langage naturel**

### **A L'initiative mixte**

L'initiative du dialogue ne doit pas être réservée exclusivement à l'un ou l'autre intervenant. Le fait que le système prenne en compte cette possibilité permet des structures de dialogue autres qu'une suite de questions-réponses.

### **B Le raisonnement**

De nombreux sous-entendus sont généralement liés à la tâche à accomplir. Dans le cadre de notre application concernant la demande de renseignements administratifs, il semble naturel de supposer, lorsque le locuteur fait une demande de double nationalité, qu'il possède la nationalité française, par exemple. En considérant que cette information existe par défaut, le dialogue n'est pas alourdi et reste efficace.

Un autre phénomène de raisonnement peut être enclenché si, par exemple, le locuteur déclare qu'il a égaré sa carte d'identité : il est évident qu'il faut alors exclure l'éventualité d'une demande de renouvellement pour péremption.

### **C La stratégie du dialogue**

Différentes stratégies peuvent être envisagées. Prenons, par exemple, celle qui consiste à compléter la requête de l'utilisateur, puis à lui fournir le minimum d'informations qui satisfassent sa demande, tout en lui laissant la possibilité de poser des questions supplémentaires. Une autre possibilité est celle qui donne toutes les informations relatives au but du locuteur (grâce à des échanges plus

longs), même si celles-ci sont redondantes ou superflues. Le choix de la stratégie à adopter doit être en harmonie avec le confort et l'efficacité qui ont été souhaités lors de la réalisation du système DIAL.

## **D Les références à l'historique**

Lors d'un dialogue, une foule de références à l'historique peut exister. Par exemple, un mot dans une phrase du locuteur peut faire référence à un autre concept dans une des phrases précédentes. Des constructions elliptiques ou anaphoriques apparaissent également.

Après avoir exposé les différents problèmes liés au langage naturel, nous allons décrire comment nous envisageons la représentation du dialogue. En effet, pour pouvoir stocker de façon adéquate et opportune les informations dans l'historique du dialogue, nous avons besoin d'une structure de données qui soit pertinente, efficace et adaptée à la structure du dialogue.

### **1.2 La représentation du dialogue**

Avant de présenter la structure du dialogue, il nous faut évoquer d'un point de vue linguistique, la dimension pragmatique qui doit être apportée aux énoncés du dialogue, le niveau sémantique devant être dépassé lorsqu'on envisage les informations contenues dans un énoncé. En effet, lorsque l'on parle, on ne dit pas seulement des choses, mais on fait également des choses. Dans un contexte théorique [AUSTIN 70], 3 sortes d'actes sont produits quand une personne prononce un énoncé :

- 1° L'acte **locutoire** : il s'agit de l'acte qui consiste à dire quelque chose.

- 2° L'acte **illocutoire** : c'est l'acte réalisé lorsqu'on dit quelque chose.

Ex : l'acte de promettre, de demander, etc.

- 3° L'acte **perlocutoire** : il s'agit de l'effet que perçoit l'interlocuteur lorsqu'on lui dit quelque chose.

Ex : la peur qui est ressentie par  
l'interlocuteur suite à un énoncé  
exprimé sous le ton de la menace.

Dans un dialogue homme-machine, la force illocutoire d'une expression est fortement réduite car la communication se situe dans un monde sincère et factuel. Nous retrouvons ici la notion de sous-langage (cfr supra), qui se traduit par un contexte limité dans lequel une application précise est développée.

De plus, les actes perlocutoires ne sont pas pris en compte, car on ne tient pas compte de l'effet perçu par le locuteur suite à un énoncé produit par la machine.

Après avoir exprimé notre souci d'appréhender, de façon pragmatique, les informations contenues dans un énoncé, développons la structure du dialogue que nous utilisons dans notre travail.

## A La structure du dialogue

Dans la littérature, diverses terminologies sont utilisées. Pour nos besoins, nous allons décomposer le dialogue en différents niveaux [ROULET 80]:

- 1° L'échange : un échange minimal est composé de deux interventions successives, mais il peut posséder une structure enchassée.

2° L'intervention : elle correspond à un tour de parole de l'un des participants.

3° L'acte de dialogue : élément minimal de ce niveau de représentation, il est caractérisé par sa fonction au sein d'une conversation.

La figure 3.1 illustre cette décomposition.

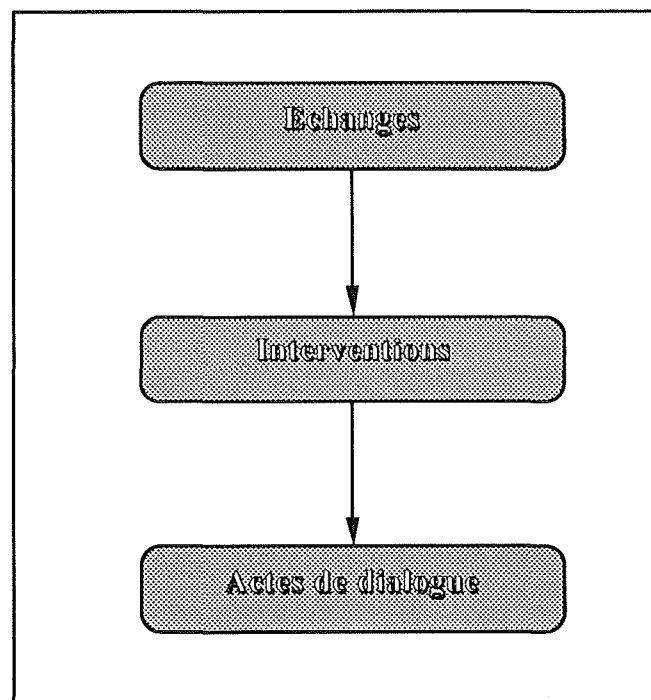


Figure 3.1 : *Structure du dialogue*

Comme nous l'avons dit auparavant, un acte de dialogue est caractérisé par sa fonction au sein d'une conversation. Il porte sur la tâche, le dialogue ou le canal de communication. Une nouvelle typologie opératoire a été définie pour que les actes soient bien adaptés au type de conversation qui est traité [ROUSSANALY 88]. Cette nouvelle typologie est illustrée à la figure 3.2.

	USAGER	SYSTEME
TACHE	Fait - Question - But	Fait - question
GESTION DU DIALOGUE	Ouverture - Clôture Demande de répétition Satisfaction Contestation	Ouverture - clôture Demande de répétition Relance
CANAL DE COMMUNICATION	Maintien Perturbation	Perturbation

Figure 3.2 : *Typologie opératoire des actes de dialogue*

Pour expliciter les différents éléments qui composent ce tableau ci-dessus, nous allons maintenant donner des exemples d'actes de dialogues. Les voici :

- **L'ouverture d'un dialogue :**

"Ici le centre de renseignements administratifs, bonjour..."

- **La clôture d'un dialogue :**

"... Au revoir"

- **La production d'informations ou fait :**

"... J'ai égaré ma carte d'identité..."

- **La question :**

"Est-ce à la suite d'une perte ou d'un vol ?"

- **Le thème ou but :**

"... Peut-on la renouveler..."

- **La demande de répétition :**

"Pouvez-vous répéter, s'il vous plaît."

- **La contestation :**

"Ce n'est pas ce que je veux."

- **La satisfaction :**

"Merci."

- **La perturbation du canal de communication :**

"Parlez plus fort..."

- **La relance :**

"Désirez-vous un autre renseignement ?"

Après avoir explicité comment nous envisageons la structure du dialogue, nous allons donner un exemple de dialogue que nous allons décomposer selon la représentation que nous utilisons (voir figure 3.1). Pour une question de clarté, cette décomposition ne prendra en compte que les deux premiers niveaux de notre hiérarchie du dialogue (figure 3.1), c'est-à-dire les échanges et les interventions.

## **B Un exemple de dialogue**

L'exemple que nous avons choisi est une simulation de demande de renseignements. Le dialogue se déroule entre un homme et une machine. Lors de la retranscription de ce dialogue, les interventions de l'expert et de l'utilisateur sont respectivement représentées par les symboles S et U suivis d'un numéro d'ordre.

S0 *Ici le centre de renseignements administratifs, bonjour*

U1 *Bonjour voilà j'ai égaré ma carte d'identité. Où peut-on la renouveler ?*

S1 *Est-ce à la suite d'une perte ou d'un vol ?*

- U2 Il s'agit certainement d'une perte
- S2 *Dans ce cas, présentez-vous à la mairie avec une déclaration de perte, deux photos d'identité et votre livret de famille ou une fiche d'état civil*
- U3 Comment obtient-on une déclaration de perte ?
- S3 *Pour une déclaration de perte, adressez-vous au commissariat*
- U4 Merci
- S4 *Désirez-vous un autre renseignement ?*
- U5 Non, merci. Au revoir
- S5 *Au revoir.*

Lorsque nous avons parlé des problèmes liés au langage naturel (voir point 1.1 du chapitre 3), nous avons évoqué le fait qu'il est intéressant que l'initiative du dialogue soit mixte, c'est-à-dire qu'elle ne soit pas la propriété exclusive d'un des deux interlocuteurs. Elle permet ainsi d'éviter un dialogue qui serait composé d'une suite de questions-réponses. Ainsi voit-on apparaître dans notre structure de dialogue des échanges imbriqués. La figure 3.3 illustre comment nous décomposons le dialogue que nous avons envisagé lors de notre exemple ci-dessus [ROUSSANALY 91].

Les symboles S et U représentent respectivement les interventions de l'expert et de l'utilisateur. Le symbole E représente l'échange. Les chiffres qui suivent ces trois symboles sont leur numéro d'ordre.

Par ailleurs, la notion d'échange est récursive. Cette récursivité est dénotée par les deux symboles ' et '' ajoutés au symbole E. On obtient donc la structure de dialogue suivante :



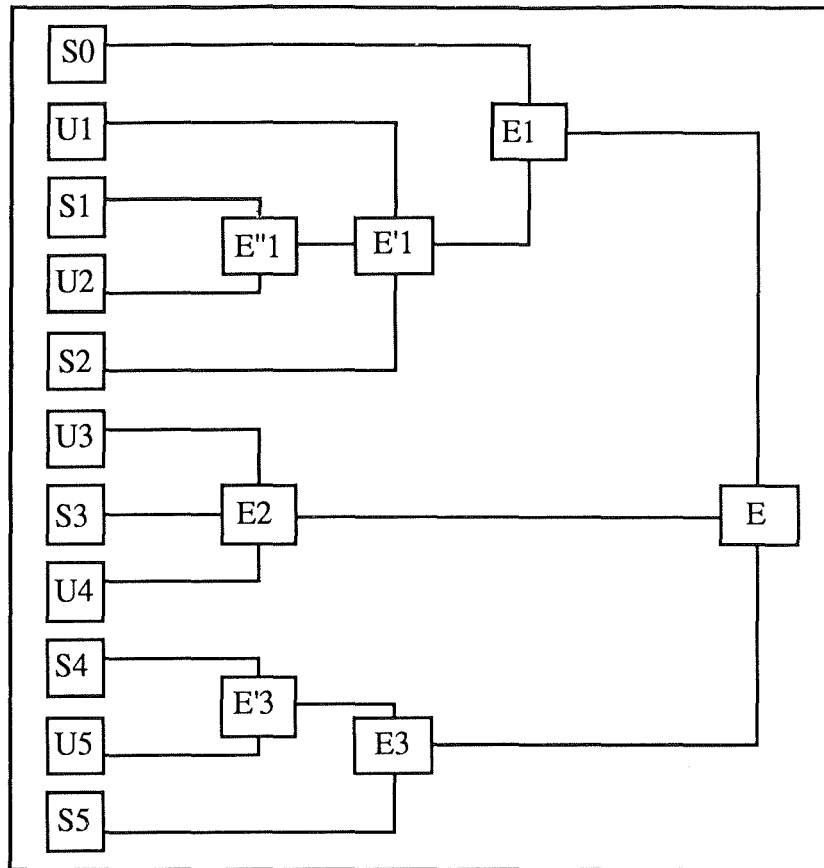


Figure 3.3 : Schéma représentatif de la structure du dialogue donné en exemple

- où
- U1, U3 et S1, S4 permet à l'un des deux interlocuteurs de prendre l'initiative du dialogue (en posant une question).
  - Le système complète d'abord la requête de l'utilisateur (S1,U2), lui fournit un minimum de renseignements (S2) et donne la possibilité au locuteur de poser des questions supplémentaires (U3,S3). On a ici une illustration de la stratégie adoptée par le système.
  - (S4, U5) représente un référence elliptique.

Voyons maintenant comment l'historique du dialogue se compose, tout en tenant compte de la structure du dialogue que nous venons d'exprimer.

## 2 L'historique du dialogue

### 2.1 Le modèle de représentation des données

Nous avons présenté plus haut l'historique du dialogue comme une structure de données à laquelle est associée une librairie de fonctions (constructeurs ou fonctions d'accès). Cette structure est décomposée en trois couches :

1. Une couche **Historique du dialogue**. Cette couche est dynamique. Elle va recevoir l'interprétation de l'énoncé courant faite par l'interpréteur. Il s'agit en fait d'une base de données évolutive en fonction du dialogue.
2. Une couche **Graphe conceptuel**. cette couche est statique et représente de manière conceptuelle les connaissances générales du monde - ici limitées à notre application.
3. Une couche **Primitives supérieures**. Elle contient les primitives prédicatives utilisées pour la représentation sémantique de l'énoncé produite par l'analyseur syntaxico-sémantique SYN-SEM.

Cette décomposition en trois couches (voir figure 3.4) est voulue par l'hétérogénéité des données et par le caractère dynamique ou non des structures manipulées. De plus amples informations à propos des éléments appartenant à ces couches seront données par la suite.

Comme nous l'avons dit auparavant, il nous faut affiner le précédent modèle de dialogue (cfr figure 3.1) pour le stockage adéquat et efficace dans la couche historique du dialogue. Les actes de dialogue sont décomposés en formules qui contiennent des ensembles de concepts individuels. Lorsqu'on fait référence à l'historique du dialogue, on recherche non pas un concept individuel, mais un ensemble de concepts individuels, ensemble qui possède certains attributs (qui seront présentés lors de la définition d'un

formalisme au point D plus bas). Ce ne sont plus des concepts qui seront reliés entre eux mais des ensembles de concepts. Cette nouvelle hiérarchie peut être représentée par la figure 3.5.

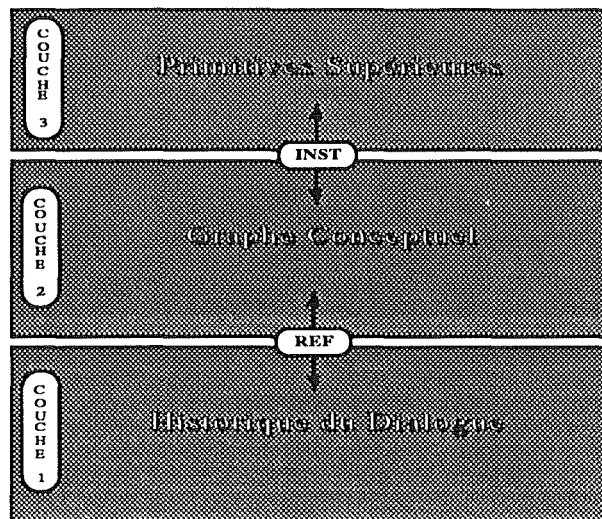


Figure 3.4 : *Décomposition de l'historique en couches*

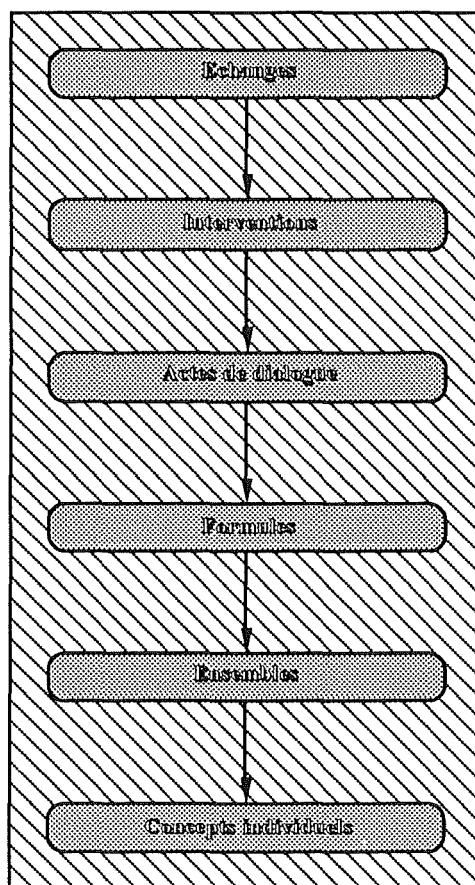


Figure 3.5 : *Structure affinée du dialogue*

Après avoir décrit la décomposition en couches de l'historique du dialogue ainsi que la nouvelle structure de dialogue que nous allons utiliser, précisons quels sont les différents types de concepts que nous allons retrouver dans notre modèle (voir figure 3.4) et dans quelle couche ceux-ci se trouvent.

## A Les types de concepts

### A.1 Les concepts génériques

Un concept générique est une entité du graphe Conceptuel (c'est-à-dire une Variable).

Ex : [PERSONNE], [LIEU].

Ce type de Concept se trouve uniquement dans le Graphe Conceptuel et peut être référencé par un ensemble d'instanciations produit par un énoncé du dialogue.

### A.2 Les concepts individuels

Un concept individuel est une instanciation d'une entité du Graphe Conceptuel (concept générique), c'est-à-dire une Constante.

Ex : "Bertrand" et "Nancy" sont respectivement des instanciations de [PERSONNE] et [LIEU].

Ce type de Concept se trouve dans l'Historique du Dialogue ainsi que dans les bases de renseignements et de données.

**N.B.** : Notion de Déictique : Certains concepts individuels sont fixes : leur existence ne dépend pas des énoncés de dialogue entre la machine et son interlocuteur. Ils donnent en quelque sorte des constantes d'environnement. Ces concepts sont appelés des Déictiques. Voici quelques exemples : "Locuteur", "Nancy", etc.

Après avoir distingué les deux types de concepts que l'on retrouve dans notre modèle de l'historique, explicitons de manière

plus précise et complète quels sont les éléments (conceptuels ou non) et les relations qui composent les différentes couches.

## B Explications des composants des différentes couches

### B.1 Couche "Primitives Supérieures"

Cette couche représente l'ensemble des primitives de prédicats associées à une grammaire de cas [DEVILLE 89]. Elles définissent en fait le modèle sémantique de SYN-SEM. La figure 3.6 donne un résumé des différentes primitives et de leurs cas associés.

PRIMITIVES (Cas,...)	Exemples
ACT_1 (Agent).....	pointer
ACT_2 (Agent, Thème).....	éditer
ACT_3 (Agent, Thème, Référent).....	associer
ECHPROD_3 (Agent, Thème, Bénéficiaire).....	donner
MVMT_1 (Agent).....	se balader
MVMT_2 (Agent, Source/Direction/Chemin).....	aller
MVMT_3 (Agent, Thème, Source/Direction/Chemin).....	mettre
COGNT_1 (Agent).....	réfléchir
COGNT_2 (Agent, Thème).....	regarder
COGNT_3 (Agent, Thème, Cognition).....	dire
STATUT_1 (Thème).....	être mineur
STATUT_2 (Thème, Référent).....	ressembler
POSSESS_2 (Bénéficiaire, Thème).....	posséder
LOCATION_2 (Thème, Lieu).....	se trouver
LOCATION_3 (Position, Thème, Lieu).....	tenir
ST_COGNT_2 (Cognition, Thème).....	savoir
PROCESS_1 (Processeur).....	fonctionner
PROCESS_2 (Effectuant, Thème).....	abîmer
ECHOBT_2 (Bénéficiaire, Thème).....	recevoir
PR_MVMT_1 (Processeur).....	tomber
PR_MVMT_2 (Processeur, Source/Direction/Chemin).....	aboutir
PR_MVMT_3 (Effectuant, Thème, Source/Dir./Chemin).....	déplacer
PR_COGNT_2 (Cognition, Thème).....	voir

Figure 3.6 : Tableau des primitives prédictives

Cette couche est donc composée d'une liste de primitives qui déterminent l'appartenance d'un concept à une classe bien précise. Par exemple, 'aller' appartient à la classe MVMT\_2 , à laquelle sont associés les cas 'agent' et 'source/direction/chemin' (s/c/d). la phrase suivante est donc possible : "Je (agent) vais (MVMT\_2) au commissariat (s/d/c)". Pour plus de détails, on pourra se reporter à la description de la composante SYN-SEM (chapitre 2) et à [DEVILLE 89]. On a donc :

Eléments : Primitives prédicatives.

EX : ECHPROD\_3, MVMT\_2, COGNT\_3, etc.

## B.2 Couche "Graphe Conceptuel"

Elle représente une base de données statiques de concepts génériques reliés par des relations de type hiérarchique [HENDRIX 79] [SOWA 84] ou casuel.

Eléments : Concepts génériques.

EX: PERSONNE, VILLE, DOCUMENT, etc.

Relations : A. Relation Hiérarchique *Sorte\_de*.

EX: *Sorte\_de* (PASSEPORT, DOCUMENT)

*Sorte\_de* (LIEU, INANIME).

B. Relations Casuelles.

EX: *Objet* (DEMANDER, DOCUMENT)

*Agent* (FOURNIR, LOCUTEUR).

La figure 3.7 illustre par un exemple les relations qui existent entre différents concepts du Graphe conceptuel. Dans cet exemple, l'interpréteur stocke l'interprétation de l'énoncé courant "*Je dois fournir une carte d'identité à la mairie*" dans la couche historique du dialogue. Cette interprétation comporte trois informations qui sont :

1. L'agent de "fournir" est "locuteur"
2. Le thème de "fournir" est "carte d'identité"
3. Le bénéficiaire de "fournir" est "mairie"

"fournir", "locuteur" et "mairie" sont en fait considérés chacun comme des ensembles. Ces ensembles référencent chacun un concept générique du graphe conceptuel. Par exemple, ici, "mairie" référence le concept générique [MAIRIE] par la relation 'Ref'.

Certains concepts génériques du Graphe conceptuel appartiennent à des classes sémantiques de la couche Primitives supérieures. La relation 'Inst' dénote cette appartenance.

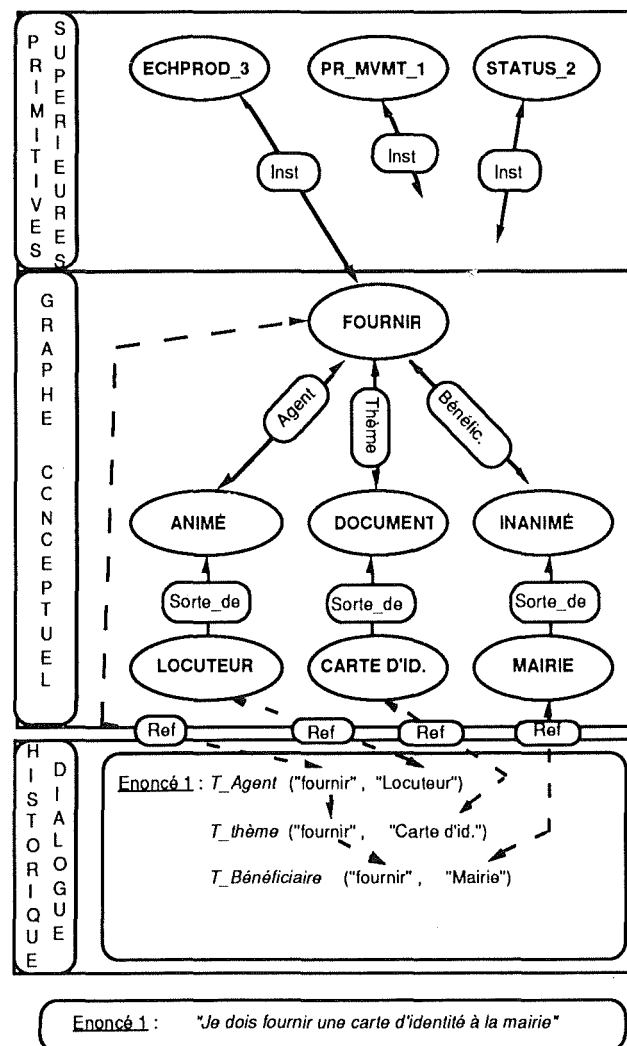


Figure 3.7 : Stockage de l'énoncé " Je dois fournir une carte d'identité à la mairie "

### B.3 Couche "Historique du Dialogue"

Elle représente une base de données évolutive en fonction du dialogue. Elle reçoit l'interprétation de l'énoncé courant faite par l'interpréteur.

Eléments : Ensembles de concepts individuels.

EX: {"Carte d'identité"}, {"photo"}, {"Nancy"}, etc.

Relations : Relations de type Casuel différentes de celles définies dans le Graphe Conceptuel.

EX: *T\_Objet* ({"demande"}, {"un document"})

*T\_Agent* ({"fournis"}, {"je"}).

Les relations entre les éléments de l'historique du dialogue sont illustrées dans l'exemple ci-dessus (voir figure 3.7).

## C Explications des différentes relations inter-couches

### C.1 Relations entre les couches Historique du dialogue et Graphe conceptuel

Le relation *Ref* établit un lien entre des Concepts génériques (G.C) et des ensembles de concepts individuels (H.D.). Ce lien permet de maintenir une certaine cohérence dans les énoncés qui sont stockés dans la couche Historique du dialogue.

EX: *Ref* (DOCUMENT , {"un document"})

*Ref* (PHOTO , {"les photos"}).



## C.2 Relations entre les couches Graphe conceptuel et Primitives supérieures

La relation *Inst* établit un lien entre des Primitives prédictives (P.S.) et des Concepts génériques (G.C.), certains concepts génériques appartenant à des classes sémantiques définies dans la couche Primitives supérieures

EX: *Inst* (ECHPROD\_3 , FOURNIR)

*Inst* (MVMT\_2, ALLER).

Après avoir passé en revue les éléments et les relations constituant des différentes couches, ainsi que les relations qui existent entre elles, nous allons illustrer par la description d'un formalisme, pourquoi nous avons affiné notre structure de dialogue (voir figure 3.5).

## D Le formalisme

Nous avons imaginé ce formalisme pour illustrer comment différents éléments de notre hiérarchie du dialogue (voir figure 3.5) sont mis en oeuvre lorsqu'un énoncé courant vient d'être interprété et est stocké dans la couche Historique du dialogue. Ce formalisme ne reprend que les deux premiers niveaux de cette hiérarchie, à savoir les ensembles et les concepts individuels ainsi que les relations internes et externes de la couche Historique du dialogue. Il va également nous permettre de définir certaines caractéristiques que des ensembles de concepts individuels doivent avoir pour atteindre et conserver la dimension pragmatique d'un énoncé du dialogue.

## D.1 Les ensembles de concepts

[ **CONCEPT** : { \* } ; **Cond**(n) ; **TQuant** ; **Rel**{ \* } ]

Où <b>CONCEPT</b>	est un Concept générique (ex : PERSONNE).
:	note une référence entre le <b>CONCEPT</b> et { * }.
{ * }	est l'ensemble des instanciations.
<b>Cond</b> (n)	Donne une condition sur n qui est le cardinal de { * } (ex : n >= 2).
<b>TQuant</b>	indique le Type du Quantificateur, c'est-à-dire : DE, IN, CA (Voir point F).
<b>Rel</b> { * }	indique la relation qui existe entre les différents éléments de { * } : DISTR, COLLT, DISJT, RESPT [SOWA 84].

Remarques : **Rel**{ \* } est omise quand le cardinal de { \* } est 0 ou 1.  
**Cond**(n) et **TQuant** sont omis (ou facultatif) si les éléments de { \* } sont tous identifiés et instanciés ("concept individuel").

Donnons maintenant quelques exemples qui illustrent comment certaines nuances de la langue française peuvent être prises en compte par l'utilisation de ce formalisme.

Exemples :

• "Un renseignement"

[ RENS : { \* } ; n=1 ; IN]

• "Le renseignement"

[ RENS : { \* } ; n=1 ; DE] ou [ RENS : { \* } ; n=1 ; CA]

- "Les renseignements"

[RENS : { \* } ; n >= 2 ; CA ; COLLT]

ou

[RENS : { \* } ; n >= 2 ; DE ; COLLT]

- "Jacques et Jean"

[PERSONNE : { "Jacques" , "Jean" } ; COLLT]

Pour d'autres exemples se reporter au point H ci-dessous.

Lors d'un dialogue homme-machine, les énoncés du locuteur et de la machine produisent des ensembles de concepts individuels. Ces ensembles sont stockés dans la couche Historique du dialogue. Des relations casuelles établissent des liens entre eux.

## D.2 Les relations de type casuel

Les ensembles de concepts individuels sont liés entre eux par des relations de type casuel, différentes de celles utilisées dans le Graphe Conceptuel. En effet, ces dernières associent des concepts génériques et non des ensembles de concepts individuels (comme dans la couche historique du Dialogue).

[CONCEPT1] -  $\rightarrow$  ( T\_RelCasuelle )  $\rightarrow$  [CONCEPT2]  
 -  $\rightarrow$  ( T\_RelCasuelle )  $\rightarrow$  [CONCEPT3]

Où T\_RelCasuelle est une Relation de type Casuel.

(ex : T\_OBJET, T\_AGENT)

CONCEPT1

CONCEPT2

sont des ensembles de concepts.

CONCEPT3

Illustrons maintenant par des exemples les relations de type casuel entre différents ensembles de concepts.

Exemples :

*"Mon père et moi devons nous rendre ensemble à la Mairie".*

[SE RENDRE] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Mon père", "moi"} ; DE ;  
COLLT]

-  $\Rightarrow$  (T\_LIEU)  $\Rightarrow$  [ADMINISTRATION : {"Mairie"} ; DE]

*"La carte d'identité et la carte de séjour coûtent respectivement 115 et 120 Francs".*

[COUTER] -  $\Rightarrow$  (T\_THEME)  $\Rightarrow$  [DOCUMENT : {"c.i.", "c.s."} ; CA ; RESPT]

-  $\Rightarrow$  (T\_REFERENT)  $\Rightarrow$  [FRANCS : {"115", "120"} ; DE ; RESPT]

Il existe également d'autres relations que les relations de type casuel. Ces relations permettent d'inclure distinctement ou pas un ensemble dans un autre, ou bien d'égaliser deux ensembles.

### D.3 Les relations d'ordre

Les énoncés d'un dialogue génèrent des ensembles d'instanciations en référence à des concepts génériques du Graphe Conceptuel. Plusieurs de ces ensembles peuvent référencer un même concept générique (Voir point E). Pour obtenir et garder une cohérence tout au long du dialogue, il faut donc les gérer. Pour cela, nous allons utiliser trois relations .

[CONCEPT1] -  $\rightarrow$  ( Rel\_Ordre)  $\rightarrow$  [CONCEPT2]  
-  $\rightarrow$  ( Rel\_ordre)  $\rightarrow$  [CONCEPT3]

Où **Rel\_Ordre** est une Relation d'ordre  
(voir explications ci-dessous).

**CONCEPT1**

**CONCEPT2** sont des ensembles de concepts.

**CONCEPT3**

Définissons de manière précise les relations d'ordre que nous allons utiliser lors de la gestion des ensembles de concepts stockés dans la couche Historique du dialogue.

Soit E1, E2 et E3 des ensembles d'instanciations de concepts génériques.

#### D.3.1 Relation d'égalité

E1 égal E2 ssi les éléments de E1 sont identiques aux éléments de E2.

Ex : "Je dois fournir des photos d'identité ? "

"Ces photos doivent être en noir et blanc ? "

#### D.3.2 Relation d'inclusion

E1 inclus E2 ssi E1 est un sous-ensemble de E2 et  $E1 \cap E2 = E1$   
et  $E2 - E1 \neq \emptyset$

Ex : "Je dois fournir des photos d'identité ? "

"Certaines photos doivent être en noir et blanc ? "

### D.3.3 Relation d'inclusion distincte

E1 inclusdist E2 ssi E1 et E2 sont des sous-ensembles de E3  
et  $E1 \cap E2 = \emptyset$

Ex : "Je dois fournir des photos d'identité ? "

"Deux de ces photos doivent être en noir et blanc tandis que les autres peuvent être en couleur."

Après avoir défini les ensembles de concepts ainsi que les relations de type casuel et d'ordre, précisons un peu ce que nous entendons par la relation de référence entre la couche Historique du dialogue et la couche Graphe conceptuel.

## E Les références

Une référence entre la couche "Graphe Conceptuel" (G.C.) et la couche "Historique du Dialogue" (H.D.) est une référence entre un Concept générique et un ensemble d'instanciations (voir figure 3.8).

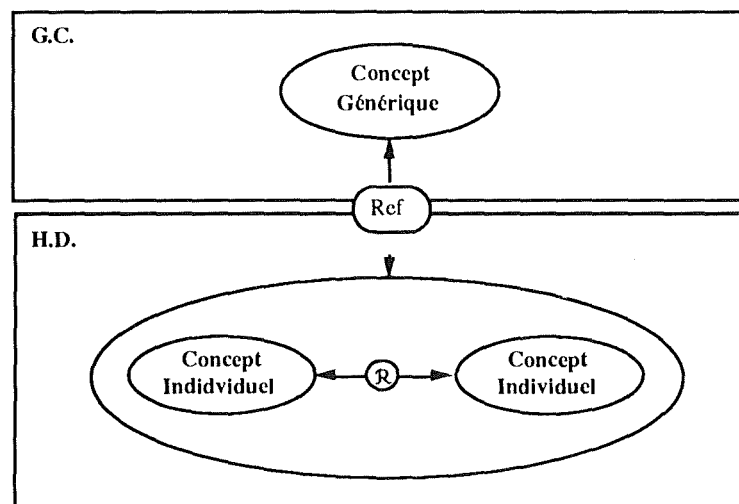


Figure 3.8 : Référence entre un ensemble et un concept générique

Un énoncé dans un dialogue produit des ensembles d'instanciations dans la partie dynamique de l'Histoire du Dialogue. Ces ensembles peuvent être de deux types : soit leurs éléments référencent le même concept générique, soit leurs éléments référencent des concepts génériques différents. De cette distinction découlent deux représentations dynamiques différentes :

### E.1 Références à un même concept générique

Les concepts individuels référencent le même concept générique et sont reliés entre eux par une relation  $\mathcal{R}$  (Voir partie G).

Ex : "Deux photos"

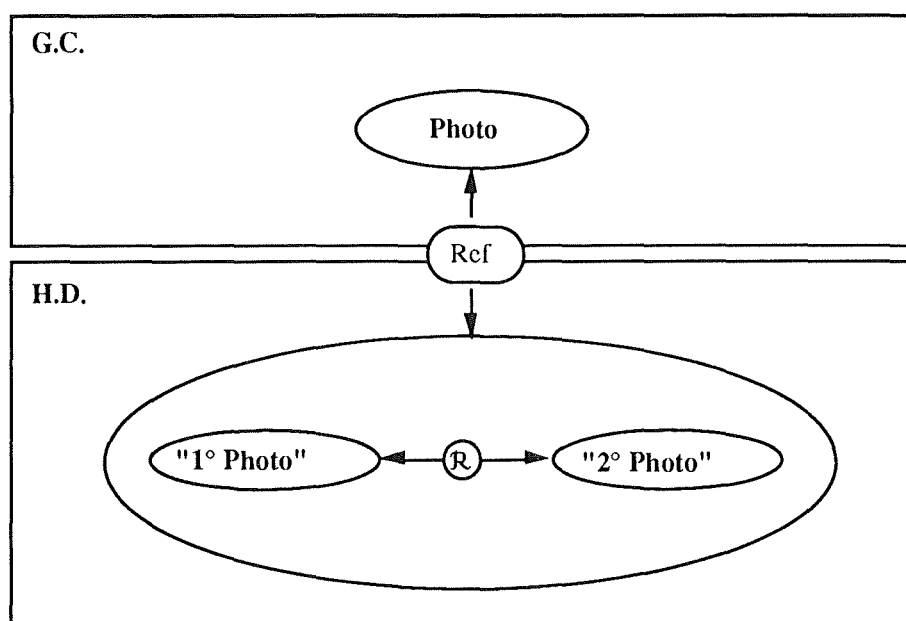


Figure 3.9 : Référence entre un ensemble de concepts individuels et un même concept générique

Chacune des deux photos pourra dans un énoncé futur être référencée, c'est-à-dire que, par exemple, un ensemble des instanciations concernant "une photo" produit par un énoncé ultérieur pourra alors être mis en relation avec l'ensemble d'instanciation comportant les "deux photos" et cela au moyen de relations d'ordre vues précédemment.

## E.2 Références à des concepts génériques différents

Les concepts individuels référencent des concepts génériques différents et sont reliés entre eux par une relation  $\mathcal{R}$  (Voir partie G). Ils référencent ensemble un concept générique qui leur est commun, et individuellement, leur propre concept générique.

La représentation ci-dessous (Voir figure 3.10), au moyen de la relation "inclus" entre les différents ensembles, rend possible la référence ultérieure à un des deux concepts séparément (soit "Carte d'identité", soit "Carte de séjour").

Ex : "La Carte d'identité et la Carte de séjour"

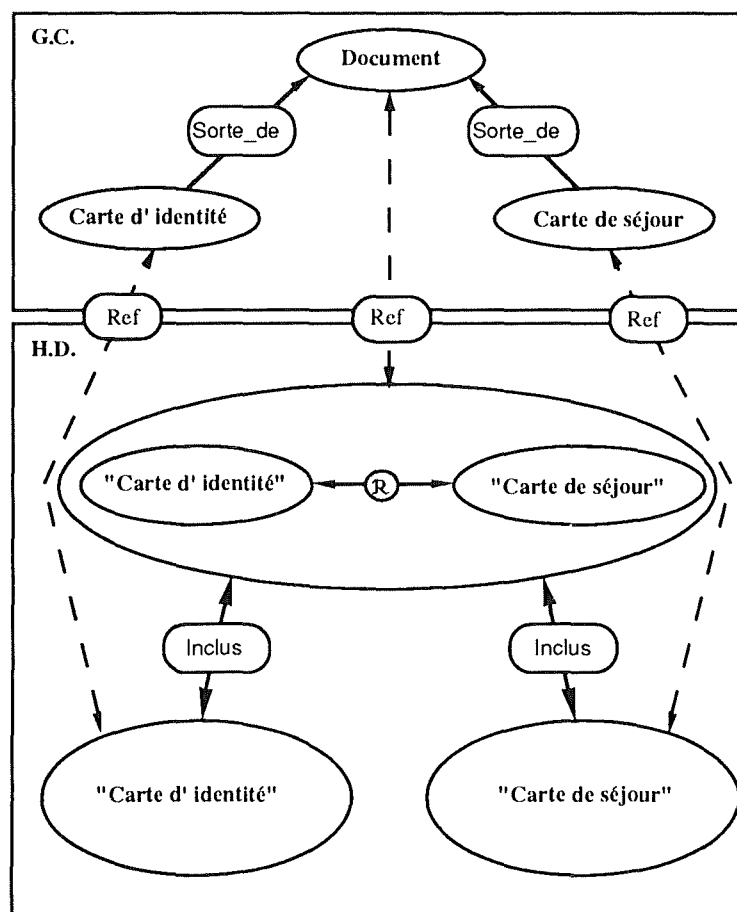


Figure 3.10 : Références entre un ensemble de concepts individuels et des concepts génériques différents



Après avoir décrit la gestion des références entre le graphe conceptuel et l'historique du dialogue, nous allons définir la notion de quantification que nous avons introduite dans ce formalisme (voir partie D.1) et expliquer pourquoi nous considérons cette notion comme importante.

## F Les quantificateurs

Les quantificateurs qui apparaissent dans des groupes nominaux telles que "*Les photos*", "*Certains renseignements*", "*une demande de double nationalité*"... etc., apportent une certaine nuance aux énoncés de dialogue et doivent donc être pris en compte lors de l'interprétation de ces énoncés.

Une discussion théorique sur le rôle des quantificateurs en langage naturel dépasse évidemment le cadre de notre travail. Néanmoins, l'interprétation d'énoncés dans un système de dialogue fait nécessairement appel à la notion de quantification. Dès lors, nous proposons une classification opératoire des quantificateurs dans le sous-langage de l'application.

Le tableau ci-dessous (figure 3.11) n'est pas exhaustif et reprend à titre d'exemple une partie des déterminants du nom, à savoir les articles. Ces déterminants sont divisés en trois classes :

1. La classe **DETERMINE** regroupe les déterminants qui qualifient un ensemble d'objets déterminés, ces déterminants pouvant donc avoir une valeur anaphorique.
2. La classe **INDETERMINE** regroupe les déterminants qui qualifient un ensemble d'objets indéterminés.
3. La classe **CATEGORIE** regroupe les déterminants qui qualifient de manière générique un ensemble d'objets.

Notons que ces quantificateurs sont considérés ici comme étant les attributs possibles d'un ensemble de concepts (dans la couche

historique du dialogue) et que cette classification doit être supplée par une condition sur l'ensemble des instances qu'ils produisent pour obtenir leur "évaluation" correcte.

	DETERMINANTS	EXEMPLES
D E T E R M I N E	<ul style="list-style-type: none"> <li>- Numéral → "Cinq photos sont nécessaires"</li> <li>- Article défini singulier → "La photo (en question) doit être récente"</li> <li>- Article défini pluriel → "Les photos (que je vous demande) doivent être récentes"</li> </ul>	DE
I N D E T E R M I N E	<ul style="list-style-type: none"> <li>- Article indéfini singulier → "Une photo d'identité (quelconque) doit avoir les dimensions suivantes : ....."</li> <li>- Article indéfini pluriel → "Vous devez fournir des photos"</li> </ul>	IN
C A T E G O R I E	<ul style="list-style-type: none"> <li>- Article défini singulier → "La photo couleur est (en général) composée de 256 couleurs"</li> <li>- Article défini pluriel → "Les photos sont en couleur ou en noir et blanc (en général)"</li> <li>- Article indéfini singulier → "Une photo couleur est composée (en général) de 256 couleurs"</li> </ul>	CA

Figure 3.11 : Classification des quantificateurs

Comme nous venons de le dire, la notion de quantification est un des types d'attribut d'un ensemble de concepts individuels. Un autre type d'attribut d'un ensemble est celui qui prend en compte les relations existant entre les éléments d'un ensemble d'instanciations dans la couche historique du dialogue (voir partie D.1 ci-dessus).

## G Les relations entre les Instanciations d'un Ensemble

Comme nous l'avons dit auparavant, les références entre le Graphe Conceptuel et l'Historique du Dialogue se font entre un concept générique ( $\in$  G.C.) et un ensemble d'instanciations ( $\in$  H.D.) (Voir figure 3.8).

Il existe entre les éléments de cet ensemble différentes Relations  $\mathfrak{R}$  qui déterminent leur participation aux relations casuelles reliant les divers concepts génériques.

Ces Relations  $\mathfrak{R}$  sont au nombre de quatre :

- **DISTRIBUTIVE** : Chaque élément de l'ensemble satisfait à la relation casuelle, mais ils le font séparément.

*Ex : "Mon père et moi devons chacun remplir le formulaire".*

[REEMPLIR] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Mon père", "moi"} ; DE ;  
DISTR]

-  $\Rightarrow$  (T\_OBJET)  $\Rightarrow$  [FORMULAIRE : {\*} ; n=1 ; DE ]

- **COLLECTIVE** : Chaque élément de l'ensemble participe de manière collective à la relation casuelle.

*Ex : "Mon père et moi devons nous rendre ensemble à la Mairie".*

[SE RENDRE] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Mon père", "moi"} ; DE ;  
COLLT]

-  $\Rightarrow$  (T\_LIEU)  $\Rightarrow$  [ADMINISTRATION : {"Mairie"} ; DE]

- **RESPECTIVE** : Les éléments de l'ensemble sont ordonnés et participent de façon respective à la relation casuelle associée à un autre ensemble ordonné d'instanciations.

*Ex : "La carte d'identité et la carte de séjour coûtent respectivement 115 et 120 Francs".*

[COUTER] -  $\Rightarrow$  (T\_THEME)  $\Rightarrow$  [DOCUMENT : {"c.i.", "c.s."} ; CA ; RESPT ]

-  $\Rightarrow$  (T\_REFERENT)  $\Rightarrow$  [FRANCS : {"115", "120"} ; DE ; RESPT ]

- **DISJONCTIVE** : Un élément de l'ensemble participe à la relation casuelle, mais il n'est pas connu.

*Ex : "Je me trouve soit à Nancy, soit à Paris".*

[SE TROUVER] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [LOCUTEUR : {"je"} ; DE]

-  $\Rightarrow$  (T\_LIEU)  $\Rightarrow$  [VILLE : {"Nancy", "Paris"} ; DE ; DISJT ]

Ces relations entre les concepts individuels permettent au raisonneur d'établir correctement des déductions (cfr chapitre 4).

Donnons maintenant des exemples qui illustrent les différentes notions que nous venons de définir.

## H Des exemples

[RENS] est le concept générique de Renseignement.

- *"Un renseignement"*

[RENS : {\*} ; n=1 ; DE]

- *"Le renseignement"*

[ RENS : { \* } ; n=1 ; DE] ou [ RENS : { \* } ; n=1 ; CA]

- *"Les renseignements"*

[ RENS : { \* } ; n >= 2 ; CA ; COLLT] ou [ RENS : { \* } ; n >= 2 ; DE ; COLLT]

- *"Tous les renseignements"*

[ RENS : { \* } ; n >= 2 ; DE ; COLLT]

- *"Certains renseignements"*

[ RENS : { \* } ; n >= 2 ; NI ; COLLT]

- *"Quelques renseignements"*

[ RENS : { \* } ; n >= 2 ; NI ; COLLT ]

- *"Aucun renseignement"*

[ RENS : { \* } ; n=0 ; DE ]

- *"Cinq renseignements"*

[ RENS : { \* } ; n=5 ; DE ; COLLT ]

- *"Ce renseignement"*

[ RENS : { \* } ; n=1 ; DE ]

- *"Les deux renseignements"*

[ RENS : { \* } ; n=2 ; DE ; COLLT ]

- *"Chaque renseignement"*

[ RENS : { \* } ; n > 0 ; DE]

- *"Les cinq renseignements"*

[ RENS : { \* } ; n=5 ; DE ; COLLT]

- *"Plusieurs renseignements"*

[ RENS : { \* } ; n >= 2 ; NI ; COLLT ]

Les nombreux exemples ci-dessus nous montrent bien que le formalisme que nous avons défini essaie de prendre en compte différentes nuances apportées par la langue française à un seul concept (renseignement) lorsque ce dernier est utilisé avec différents adjectifs ou articles.

Après avoir défini la structure de données que nous allons utiliser dans la couche Historique du dialogue, il nous faut décrire quelles sont les différentes fonctions qui composent la librairie mise à la disposition des modules de traitements de DIALOG. Grâce à cette librairie, le raisonneur, l'interpréteur et le dialogueur pourront réaliser les travaux qui leur incombent.

## 2.2 La gestion de l'historique de dialogue

L'Histoire peut être considéré comme un Noyau : La partie centrale est la représentation dynamique de toutes les données tandis que les fonctions d'accès et les constructeurs composent la couche supérieure (Voir figure 3.12). La hiérarchie de la structure de données utilisée est décrite à la figure 3.5. Dans le cadre de ce travail, nous limiterons nos recherches aux quatre premiers niveaux, c'est-à-dire les concepts individuels, les ensembles, les formules et les actes de dialogues.

Les deux premiers niveaux ont fait l'objet de recherches plus poussées tandis que les deux suivants doivent encore être développés de manière plus approfondie.

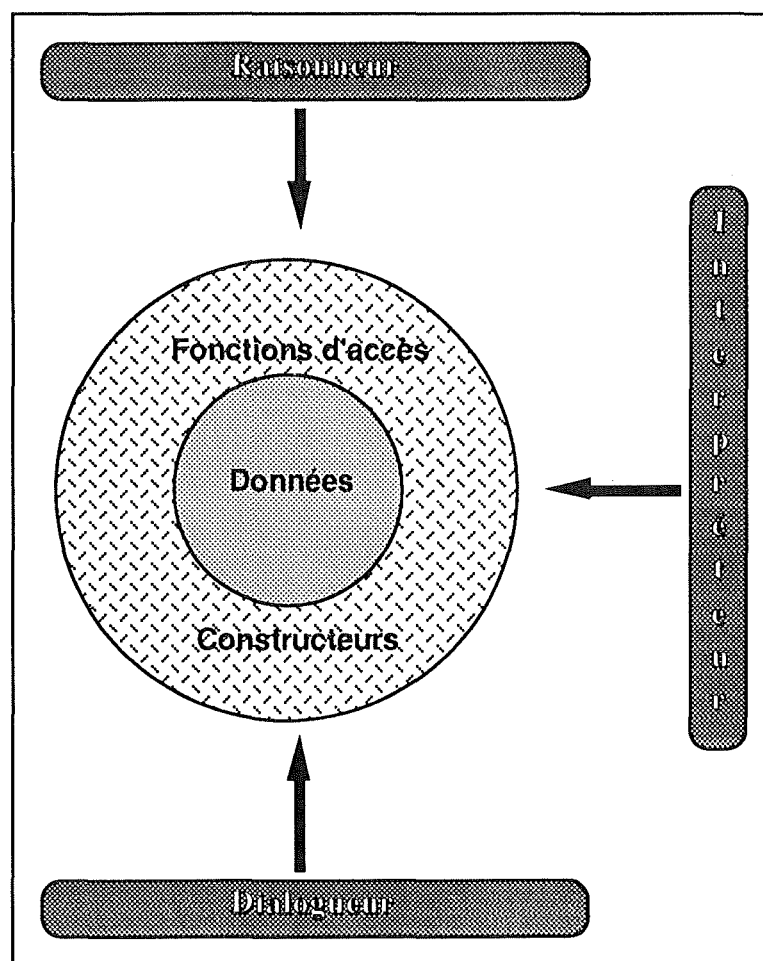


Figure 3.12 : *Gestion de l'Histoire du Dialogue.*

Détaillons quelque peu quelles seront les différentes fonctions mises à la disposition de l'interpréteur, du raisonneur et du dialogueur. Elles font l'objet de spécifications précises à l'annexe A.

## **A Les concepts individuels ou noeuds**

### Constructeurs :

1. Construire (détruire) un noeud.
2. Construire (détruire) une relation entre les noeuds.
3. Modifier un noeud.
4. Modifier une relation.

### Primitives d'accès :

1. Rechercher un noeud selon un critère.
2. Rechercher un type de relation.

## **B Les ensembles**

### Constructeurs :

1. Construire (détruire) un ensemble.
2. Construire (détruire) une référence entre un ensemble et un concept générique.
3. Construire (détruire) une relation entre deux ensembles.

### Primitives d'accès :

1. Rechercher un ensemble selon un critère.
2. Rechercher un type de relation.
3. Recherche d'une référence dans le Graphe Conceptuel.



## C Les formules

### Constructeurs :

1. Construire (détruire) une formule.
2. Construire (détruire) une relation entre des formules.
3. Construire (détruire) un point de synchro.

### Primitives d'accès :

## D Les actes de dialogue

### Constructeurs :

1. Construire (détruire) une liste vide.
2. Ajouter (enlever) un acte de langage dans une liste d'actes de dialogue.

### Primitives d'accès :

1. Donner le dernier acte de langage d'une liste d'actes de dialogue.
2. Tester si une liste est vide.

Différentes fonctions qui ont été évoquées ci-dessus sont reprises dans des exemples (schémas) illustrant le fonctionnement de l'application que nous avons réalisée pour ce travail. On pourra se référer à l'annexe E pour obtenir plus de détails.

## **Chapitre 4**

# **Les résultats et perspectives**

## **1 Les limitations**

Après avoir décrit la représentation des données dans l'historique du dialogue ainsi que les primitives permettant la gestion de ces dernières, nous nous sommes volontairement limités dans la conception et la réalisation d'une application montrant les grandes idées qui ont été dégagées lors de ce travail. Pourquoi cette limitation ? Il nous est apparu que, dans la nouvelle structure de dialogue (cfr figure 3.5) implantée dans notre modèle de l'historique (cfr figure 3.4), la notion d'ensemble est tout à fait cruciale tant par l'originalité qu'elle apporte que par sa souplesse lors de modifications ultérieures. C'est pourquoi seuls les fonctions d'accès et les constructeurs au niveau des concepts individuels et des ensembles ont été réalisés pour cette application.

## **2 L'élaboration d'une application**

### **2.1 Les explications**

L'application a été réalisée en langage LISP (plus précisément en Common Lisp). Le choix du langage de programmation a été déterminé par les facilités d'implémentation et la philosophie de ce langage vis-à-vis des structures de données choisies (cfr annexe A).

La structure physique de l'application est composée de fichiers de données, de fichiers Lisp comprenant les primitives de traitements et de fichiers de résultats. La figure 4.1 illustre cette architecture d'un point de vue général. Pour plus de précision, on se reportera à l'annexe C.

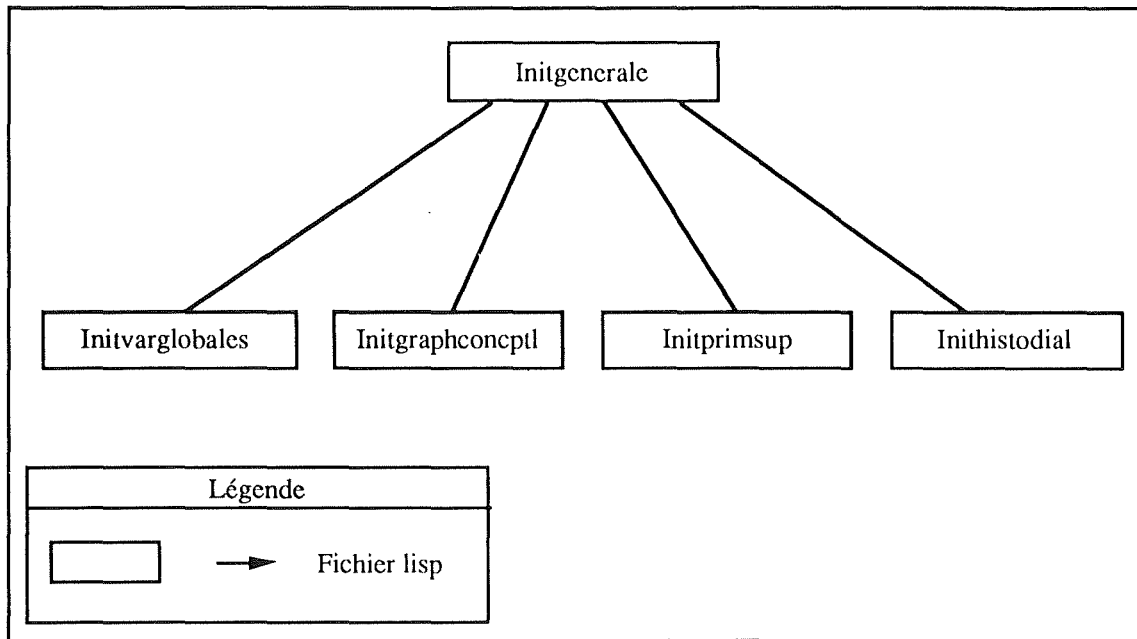


Figure 4.1 : *Architecture physique de notre application*

Un concept individuel ou générique est représenté physiquement par un noeud, c'est-à-dire un couple dont le premier élément est le numéro du noeud et le deuxième le nom du concept. Par exemple, 'chat' est représenté par '(23 chat)'. Un ensemble est représenté par un quintuplé formé successivement par un numéro d'ensemble, une condition sur l'ensemble, un type de quantificateur, un type de relation entre les éléments de l'ensemble et, pour terminer, la liste des éléments de l'ensemble. Prenons, par exemple, l'ensemble '{document papier}' ; il sera représenté, après interprétation du contexte, par '[34 (N = 2) DE COLLT (document papier)]' et stocké dans la couche historique du dialogue.

Les différentes structures de données qui sont utilisées ici sont reprises et expliquées à l'annexe A.

Nous allons maintenant illustrer comment fonctionne une fonction de notre librairie (voir figure 4.1) et quelles sont les répercussions sur la structure de données de la couche historique du dialogue.

## 2.2 Des exemples

Donnons ici deux exemples qui utilisent la fonction '**creens**' (fonction de la librairie que nous avons définie pour manipuler les données de l'historique), laquelle permet de créer un ensemble de concepts individuels dans la couche historique du dialogue. Les arguments qu'il faut donner à cette fonction sont, successivement, le type de quantificateur, le type de relation entre les éléments de l'ensemble et la liste des éléments de cet ensemble.

Cette fonction crée donc un ensemble avec les arguments qu'on lui a passés comme paramètres et lui attribue un numéro (cfr supra et annexe).

L'application résout également les références entre les ensembles de concepts individuels et les concepts génériques et gère les relations entre ces ensembles.

Il faut également noter que les deux exemples qui sont proposés ici sont successifs, c'est-à-dire que les résultats obtenus après l'application de la primitive dans le premier exemple sont également repris dans le deuxième.

Les résultats de l'application de '**creens**' sur l'ensemble des données sont mis en gras dans chacun des deux exemples.

Une liste d'ensembles existe déjà avant l'utilisation de la primitive '**creens**'. Il s'agit d'ensembles représentant des déictiques, c'est-à-dire des constantes d'environnement (voir supra et annexe D).

### Exemple n°1

On veut créer un ensemble contenant des concepts individuels 'ville' et 'nancy'. La relation entre ces éléments est de type collectif (COLLT) et le quantificateur catégorique CA est associé à l'ensemble devant être créé.

(creens 'ca 'collt '(ville nancy))

Voici la liste des ensembles :

-----  
**Ens n 1 : [216 (N = 1) CA NIL (NANCY)]**  
**Ens n 2 : [214 (N = 1) CA NIL (VILLE)]**  
**Ens n 3 : [210 (N = 2) CA COLLT (VILLE NANCY)]**  
Ens n 4 : [194 (N = 1) DE NIL (JEAN)]  
Ens n 5 : [195 (N = 1) DE NIL (MARIE)]  
Ens n 6 : [196 (N = 1) DE NIL (PIERRE)]  
Ens n 7 : [197 (N = 1) DE NIL (PAUL)]  
Ens n 8 : [198 (N = 1) DE NIL (DOCUMENT)]  
Ens n 9 : [199 (N = 1) DE NIL (ALLER)]  
Ens n 10 : [200 (N = 1) DE NIL (LOCUTEUR)]  
Ens n 11 : [201 (N = 1) DE NIL (CRI)]

Trois ensembles ont été créés, car les deux concepts individuels de l'ensemble {ville nancy} réfèrent des concepts génériques différents, c'est-à-dire que {ville} référence le concept générique [VILLE] tandis que {nancy} référence [NANCY]. Ces trois ensembles sont donc {ville}, {nancy} et {ville nancy}, ce dernier ensemble faisant référence à [VILLE], qui est le concept commun entre [VILLE] et [NANCY]. Les autres ensembles sont les déictiques (voir supra).

Réf : Ensemble : Concept générique

-----  
**217 : [216 (N = 1) CA NIL (NANCY)] : NANCY**  
**215 : [214 (N = 1) CA NIL (VILLE)] : VILLE**  
**213 : [210 (N = 2) CA COLLT (VILLE NANCY)] : VILLE**  
202 : [194 (N = 1) DE NIL (JEAN)] : PERSONNE  
203 : [195 (N = 1) DE NIL (MARIE)] : PERSONNE  
204 : [196 (N = 1) DE NIL (PIERRE)] : PERSONNE  
205 : [197 (N = 1) DE NIL (PAUL)] : PERSONNE  
206 : [198 (N = 1) DE NIL (DOCUMENT)] : DOCUMENT  
207 : [199 (N = 1) DE NIL (ALLER)] : ALLER  
208 : [200 (N = 1) DE NIL (LOCUTEUR)] : LOCUTEUR  
209 : [201 (N = 1) DE NIL (CRI)] : INANIME

Des relations entre les trois nouveaux ensembles doivent également être créées. Elles sont de type 'inclus'. {ville} et {nancy} sont inclus dans {ville nancy}. {ville} et {nancy} peuvent donc être référencés ultérieurement de façon séparée.

Voici la liste des relations entre les ensembles :

-----

[216 (N = 1) CA NIL (NANCY)] => (INCLUS) => [210 (N = 2)  
CA COLLT (VILLE NANCY)]  
[214 (N = 1) CA NIL (VILLE)] => (INCLUS) => [210 (N = 2)  
CA COLLT (VILLE NANCY)]

### Exemple n°2

On veut créer un ensemble contenant des concepts individuels 'visa' et 'carte\_ss'. La relation entre ces éléments est de type collectif (COLLT) et le quantificateur indéterminé IN est associé à l'ensemble devant être créé.

(creens 'in 'collt '(visa carte\_ss))

Voici la liste des ensembles :

-----

Ens n 1 : [224 (N = 1) IN NIL (CARTE\_SS)]  
Ens n 2 : [222 (N = 1) IN NIL (VISA)]  
Ens n 3 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
Ens n 4 : [216 (N = 1) CA NIL (NANCY)]  
Ens n 5 : [214 (N = 1) CA NIL (VILLE)]  
Ens n 6 : [210 (N = 2) CA COLLT (VILLE NANCY)]  
Ens n 7 : [194 (N = 1) DE NIL (JEAN)]  
Ens n 8 : [195 (N = 1) DE NIL (MARIE)]  
Ens n 9 : [196 (N = 1) DE NIL (PIERRE)]  
Ens n 10 : [197 (N = 1) DE NIL (PAUL)]  
Ens n 11 : [198 (N = 1) DE NIL (DOCUMENT)]  
Ens n 12 : [199 (N = 1) DE NIL (ALLER)]  
Ens n 13 : [200 (N = 1) DE NIL (LOCUTEUR)]  
Ens n 14 : [201 (N = 1) DE NIL (CRI)]

Réf : Ensemble : Concept générique

Voici la liste des relations entre les ensembles :

75

[222 (N = 1) IN NIL (VISA)] => (INCLUS) => [218 (N = 2) IN  
COLLT (VISA CARTE\_SS)]  
[216 (N = 1) CA NIL (NANCY)] => (INCLUS) => [210 (N = 2) CA  
COLLT (VILLE NANCY)]  
[214 (N = 1) CA NIL (VILLE)] => (INCLUS) => [210 (N = 2) CA  
COLLT (VILLE NANCY)]

D'autres exemples sont proposés en annexe D.

### 3 Les perspectives

Après avoir donné la structure des données de l'historique du dialogue et explicité la notion d'ensemble -qui est importante dans la représentation du modèle de cet historique-, il nous faut mettre l'accent sur les possibilités qu'offre cette notion pour un autre module interne de traitement que l'interpréteur. Le raisonneur, en effet, grâce aux attributs donnés aux ensembles de concepts individuels, peut effectuer certaines déductions. Pour mieux comprendre, prenons un exemple :

- l'interpréteur reçoit de SYN-SEM la structure syntaxico-sémantique de l'énoncé : "Mon père et moi devons chacun remplir un formulaire"
- Il l'interprète selon le contexte et stocke le résultat dans l'historique. Ce résultat est :

[REEMPLIR] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Mon père", "Moi"} ; DE ;  
DISTR ]

-  $\Rightarrow$  (T\_THEME)  $\Rightarrow$  [FORMULAIRE : {\*} ; n = 1 ; CA ]

- Le raisonneur peut alors effectuer des déductions sur l'énoncé stocké dans l'historique. Il peut le décomposer en plusieurs formules. Le résultat précédent est donc décomposé comme suit :

[REEMPLIR] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Mon père"} ; DE ]

-  $\Rightarrow$  (T\_THEME)  $\Rightarrow$  [FORMULAIRE : {\*} ; n = 1 ; DE ]



[REEMPLIR] -  $\Rightarrow$  (T\_AGENT)  $\Rightarrow$  [PERSONNE : {"Moi"} ; DE ]

-  $\Rightarrow$  (T\_THEME)  $\Rightarrow$  [FORMULAIRE : {\*} ; n = 1 ; DE ]

On remarque que cette déduction s'effectue au niveau du type de relations entre les éléments d'un ensemble (voir point G dans le chapitre 3). On peut généraliser cette déduction d'un point de vue mathématique :

Soit P une primitive prédicative (par exemple, une primitive ACT\_2 )

A, A<sub>1</sub>, A<sub>2</sub> des cas 'AGENT'

T, T<sub>1</sub>, T<sub>2</sub> des cas 'THEME'

Le raisonneur, selon le type de relations entre les éléments d'un ensemble, peut établir les déductions suivantes :

P [ A , T ]

• Distributif

$P[A_1 \text{ et } A_2, T] \rightarrow P[A_1, T] \text{ et } P[A_2, T]$

$P[(A_1, A_2), (T_1, T_2)] \rightarrow P[A_1, (T_1, T_2)]$   
 $\text{et}$   
 $P[A_2, (T_1, T_2)]$

• Collectif

$P[A_1 \text{ et } A_2, T] \rightarrow P[A_1 \text{ et } A_2, T]$

• Respectif

$P[(A_1, A_2), (T_1, T_2)] \rightarrow P[A_1, T_1] \text{ et } P[A_2, T_2]$

• Disjonctif

$$P[A, [T_1, T_2]] \rightarrow P[A, T_1] \text{ ou } P[A, T_2]$$

Une autre perspective est la poursuite de nos travaux, c'est-à-dire le développement des notions de formules, d'actes de dialogues, d'interventions et d'échanges. Le lecteur pourra d'ailleurs trouver en annexe les architectures de certaines primitives (construction ou modification d'une formule). Nous n'en donnerons ici qu'un exemple, illustré par la figure 4.2.

D'autre part, la validation pratique des différentes notions que nous avons explicitées doit encore être réalisée sur différents corpus. Cette vérification a été commencée et doit être continuée.

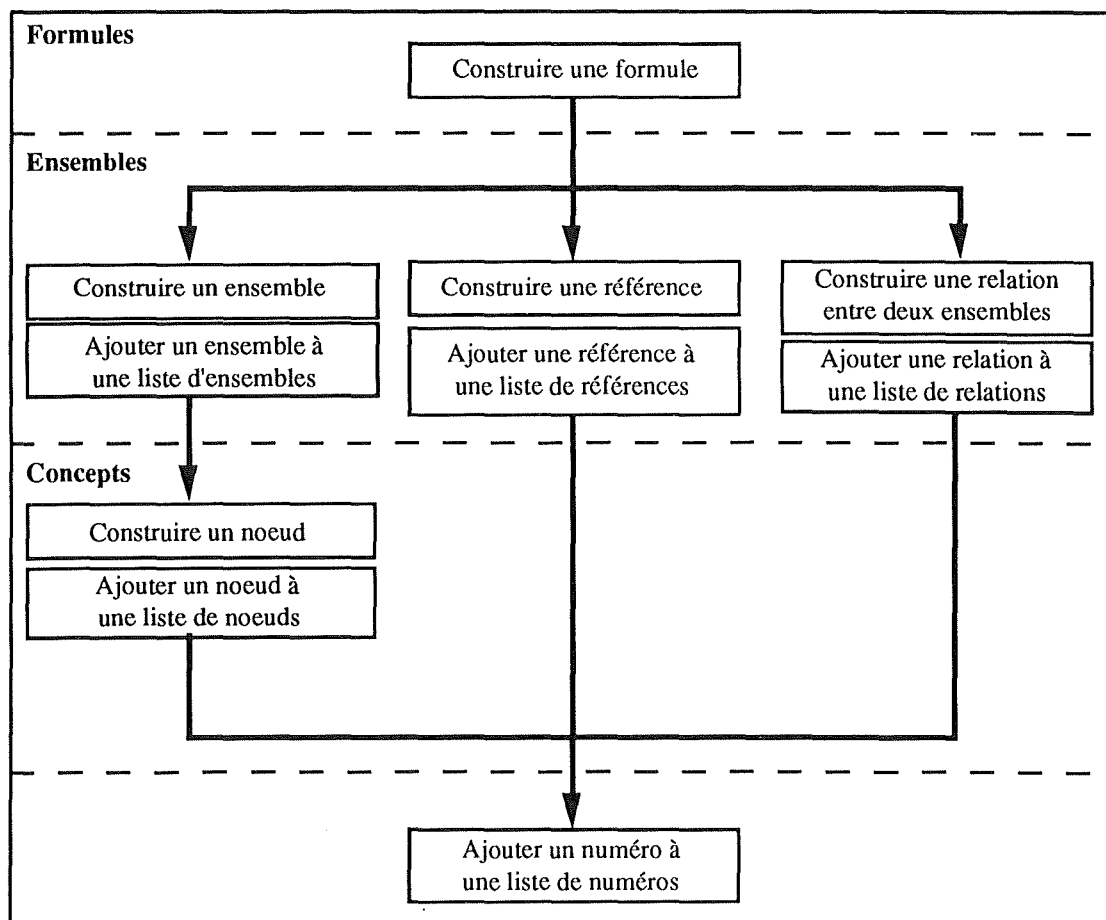


Figure 4.2 *Organigramme de la construction d'une formule*

# Conclusions

Notre objectif était d'ébaucher un modèle de représentation de l'historique du dialogue qui soit à la fois souple, efficace et bien adapté au stockage des informations traitées et interprétées par l'interpréteur. Suivant notre méthodologie de travail, nous n'avons pas voulu nous lancer à l'aventure dans un domaine très vaste et complexe. C'est pourquoi nous avons tenu tout particulièrement à asseoir nos recherches sur des bases théoriques solides [SOWA 84] et [HENDRIX 79].

Différents problèmes relatifs au langage naturel ont été évoqués au début de ce travail. Lors de l'élaboration de notre modèle, nous avons tenté d'y apporter un début de solution, en prenant en compte, dans notre structure de données, des caractéristiques susceptibles d'être utilisées par des modules de traitements internes autres que l'interpréteur, c'est-à-dire le raisonneur et le dialogueur.

La notion de quantificateur est utilisée par l'interpréteur lors de l'évaluation contextuelle de l'énoncé courant. Ce module vérifie également la cohérence de l'énoncé et résout les références employées. Le raisonneur, pour sa part, effectue des déductions grâce à l'introduction de types de relations entre les éléments d'un ensemble (voir chapitre 4).

La souplesse de notre modèle de représentation des données dans l'historique réside dans le fait que l'on ne considère plus, dans notre modèle, les concepts individuels comme les éléments principaux de référence, mais bien des ensembles de concepts individuels auxquels sont associés divers attributs indispensables au traitement pragmatique souhaité du module DIALOG.

Des problèmes subsistent encore et seront examinés de façon plus précise dans le futur. Ainsi ce travail n'a pas pris en compte l'aspect temporel des énoncés d'un dialogue et a laissé entrevoir les différents problèmes qui vont surgir lors de l'élaboration des règles d'interprétation de la composante pragmatique (prise en compte des quantificateurs).

# Bibliographie

[AUSTIN 70] J.L. Austin

*Quand dire c'est faire* , Edition du seuil, Paris, 1970.

[BOLT 76] Bolt Beranek and Newman Inc.

*Speech Understanding Systems-Final Report* , Cambridge,  
B.B.N. Report n° 3438, 1976.

[BONIN 87] J.J. Bonin

*Détection d'indices prosodiques linguistiquement significatifs*,  
Rapport  
de DEA, Nancy I, 1987.

[CARBONNEL 84] N. Carbonnel, F. Charpillet, J.P. Haton, B. Mangeol, P.  
Mousel, J.M. Pierrel, A. Roussanaly.

*Dialogue oral homme-machine : Bilan du projet MYRTILLE et perspectives* , Actes du séminaire GRECO sur le dialogue oral  
homme-machine, pp 91-122, Nancy, 1984.

[DE JONG 85] F. De Jong et H. Verkuyl

*Generalized Quantifiers : the Properness of their Strength*, Foris  
Publications, Tome 4, chapitre 2, 1985.

[DEVILLE 86] G. Deville, H. Paulussen

*A Case Grammar as an Original Linguistic Model for the  
Sémantic Representation of Utterances in a Man-Machine  
Dialog System* , Mémoire de linguistique computationnelle,  
Université d'Anvers, 1986.

[DEVILLE 87] G. Deville

*Corpus de dialogues oraux finalisés en situation réelle :  
Demande de renseignements auprès du Ministère de l'emploi  
et du travail (Cellule Action-Travail)*, Ed. G. Deville, FUNDP  
Namur, 1987.

[DEVILLE 89] G. Deville

*Modelization of Task-Oriented Utterances in a Man-Machine Dialogue System*, Thèse de Doctorat de l'Université d'Anvers, 1989.

[ERMAN 80] L.D. Erman, F. Hayes-Roth, V.R. Lesser, D.R. Reddy

*The HEARSAY II Speech Understanding System : Integrating Knowledge to Resolve Uncertainty*, Computing Surveys, volume 12, n° 2, 1980.

[FALZON 84] P. Falzon

*Les langages opératifs*, Actes du séminaire GRECO "Dialogue homme-machine à composante orale, Nancy, 11-12 octobre 1984.

[FOHR 86] D. Fohr

*APHODEX : un système expert de décodage acoustico-phonétique de la parole continue*, Thèse de doctorat, Université de nancy I, 1986.

[GRECO 85] GRECO

*Dialogue oral homme-machine en situation orientée par l'action*, Actes du 5ème congrès AFCET- RFIA.

[GRICE 75] H.P. Grice

*Logic and Conversation in Syntax and Semantic : Speech acts*, E. Cole and J.L. Morgan eds, pp 41-58, Academic Press, 1975.

[HENDRIX 79] Gary G. Hendrix

*Encoding knowledge in partitioned network*, Associative Network, N.V. Findler, ed. Academic Press, 1979.

[HORMANN 72] H. Hormann

*Introduction à la psycholinguistique. Langue et langage*, Larousse, Paris, 1972.

[KERNIGHAM 84] B.W. Kernigham et D.M. Ritchie

*Le langage C*, Masson, 1984.

- [LEVINSON 83] S. Levinson  
*Pragmatics*, Cambridge University Press, 1983.
- [LIBERMAN 70] A.M. Liberman  
*The Grammar of Speech and Language*, Cognitive psychology, volume 1, pp 301-323, 1970.
- [MASINI 89] G.MASINI, A. Napoli, D. Colnet, D. Leonard, K. Tombre  
*Les langages à objets* , InterEditions, 1989.
- [MOUSEL 90] P. Mousel  
*Syntaxe et sémantique dans un système de dialogue oral homme-machine en langage naturel*, Thèse de l'université de NancyI, 1990.
- [PIERREL 75] J.M. Pierrel  
*Contribution à la compréhension automatique du discours continu*, Thèse de troisième cycle, Université de nancy I, 1975.
- [PIERREL 87] J.M. Pierrel  
*Dialogue oral homme-machine*, Edition Hermes, Paris , 1987.
- [QUINTON 82] P. Quinton  
*Utilisation de contraintes syntaxiques pour la reconnaissance de la parole continue*, Techniques et sciences informatiques, volume 1, n° 3, 1982.
- [ROUSSANALY 86] A Roussanaly, P. Mousel, N. CARbonell, B. Mangeol, J.M. Pierrel  
*Réalisation d'un corpus de dialogues oraux. Application au renseignements administratifs*, Rapport interne au CRIN n° 86-R-083 1986.
- [ROUSSANALY 88] A. Roussanaly  
*DIAL, la composante dialogue d'un système de communication orale homme-machine finalisée en langage naturel* , thèse de l'Université de NancyI, 1988.

[ROUSSANALY 91] A. Roussanaly, J.M. Pierrel

*La composante dialogue dans un système de compréhension du dialogue oral homme-machine finalisé en langage naturel* , Article TSI, 1991.

[SOWA 84] J.F. Sowa

*Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, 1984.

[STROUSTRUP 86] B. Stroustrup

*The C++ Programming Language* , Addison-Wesley Publishing Company, 1986 .

[TOURETZKY 84] D. Touretzky

*LISP : a Gentle Introduction to Symbolic Computation*, Harper & Row Publishers New York, 1984.

[VERKUYL 87] H. Verkuyl

*Nondurative Closure of Events*, Foris Publications, Tome 8, chapitre 4, 1987.

[WINSTON 89] P.H. Winston, B. K. P. Horn

*LISP 3trd edition*, Addison-Wesley publishing company, 1989.



## Annexes :

A. Spécifications des programmes

B. Listing des programmes

C. Organigramme général de  
l'application

D. Données

E. Résultats obtenus

## **Annexe A**

### **Spécifications des programmes**

# 1. Primitives supérieures.

## 1.1 Structure des objets.

### 1.1.1 Les primitives.

Structure d'une primitive : (Numcg Nompr)

où	Numcg	entier (identifiant)
	Nompr	string

Structure d'une liste de primitives :

LINST=((Numcg Nompr)\*)

## 1.2 Spécifications internes.

### 1.2.1 Les primitives.

#### A. Constructeur.

- 1° Construire une liste de primitives.
- 2° Ajouter une primitive à une liste de primitives.
- 3° Enlever une primitive d'une liste de primitives.
- 4° Modifier une primitive.

#### B. Fonction d'accès.

- 1° Rechercher une primitive dans une liste de primitives.

## 2. Graphe Conceptuel.

### 2.1 Structure des objets.

Remarque préliminaire : le symbole \* note la répétition de l'élément après lequel il est placé.

Ex : (( NumCg NomCg )\*) donne  
(( NumCg1 NomCg1 )( NumCg2 NomCg2 )( NumCg3 NomCg3 )...)

#### 2.1.1 Les concepts génériques.

Structure d'un concept : ( NumCg NomCg )

où	NumCg	entier (identifiant)
	NomCg	string

Structure de la liste des concepts : LCG = (( NumCg NomCg )\*)

#### 2.1.2 Les relations casuelles et hiérarchique.

Structure d'une relation : (RELCASHIER NumCg1 NumCg2)

où	RELCASHIER	RELATIONS CASUELLES + ISA
	NumCg1	entier (identifiant)
	NumCg2	entier (identifiant)

Structure de la liste des relations :

LRCH = ((RELCASHIER NumCg1 NumCg2)\*)

## 2.2 Spécifications internes.

### 2.2.1 Les concepts génériques.

#### A. Constructeur.

##### 1° Construire une liste de concepts.

*Syntaxe* : **CREERLISTECP** (1)

*But* : Construire une liste de paires dont le premier élément est le numéro du concept et le deuxième le nom du concept.

*Pré* : /

*Post* : Une liste de paire est créée.

*IN* : 1 liste de noms de concepts

*OUT* : une liste de paires : ((Num Nom)\*)

##### 2° Ajouter un concept à une liste de concepts.

*Syntaxe* : **AJOUTCP** (Nomcp l)

*But* : Ajouter à une liste l une paire dont le premier élément est le numéro du concept -généré par la fonction- et le deuxième le nom du concept Nomcp donné en entrée.

*Pré* : l existe.

*Post* : Le concept de nom Nomcp est ajouté à la liste l s'il ne lui appartenait pas encore.

*IN* : Nomcp string  
l liste de concepts ((Numcp Nomcp)\*)

*OUT* : l liste de concepts : ((Numcp Nomcp)\*)

### 3° Modifier un concept.

*Syntaxe* : **MODIFCP** ( numcp NvNomcp l)

*But* : Modification d'une liste de paires .

*Pré* : /

*Post* : le Nom NomNd du noeud numéro NumNd est remplacé par NvNomNd.

*IN* :            NumNd            entier (identifiant)  
                 NvNomNd        string

*OUT* :           NumNd            entier (identifiant)  
                 NvNomNd        string

## **B. Fonction d'accès.**

### 1° Rechercher un concept dans une liste selon un critère.

*Syntaxe* : **RECHCP**(critère CG)

où le critère peut être un numéro du concept NumCg.  
un nom du concept NomCg.

*But* : Rechercher un numéro ou un nom de concept dans une liste de concepts.

*Pré* : CG existe.

*Post* : CG est inchangée.

*IN* :            NumCg            entier (identifiant)  
                 ou  
                 NomCg        string  
                 CG            liste de concepts : ((NumCg NumCg) \*)

*OUT* : • (NumCg NomCg)            si le Numéro du concept NumCG  
   ou le Nom du noeud NomCg se  
   trouve dans la liste CG.

• "Nil"                            sinon

## 2.2.2 Les relations casuelles et hiérarchique.

### 1° Construire une relation entre deux concepts.

*Syntaxe :* **CONSTRELCASHIER** ( RCH NumCg1 NumCg2 LRCH )

*But :* Construire un triplet composé comme suit : le premier élément est la relation entre les deux concepts, le deuxième et le troisième sont les numéros des concepts qui satisfont à cette relation. Cette relation est insérée dans une liste de relations de même type.

*Pré :* NumCg1, NumCg2 et LRCH existent.

*Post :* un triplet ( RCH NumCg1 NumCg2 ) est créé et ajouté à la liste LRCH.

<i>IN :</i>	RCH	relation casuelle ou hiérarchique
	NumCg1	entier (identifiant)
	NumCg2	entier (identifiant)
	LRCH	liste de relations

<i>OUT :</i>	LRCH	liste de relations
--------------	------	--------------------

## 3. Historique du Dialogue.

### 3.1 Structure des objets.

Remarque : les concepts individuels sont représentés ici par des noeuds. C'est pourquoi dans la suite de ce travail nous utiliserons uniquement le mot "noeud" pour désigner un concept individuel. Pour voir la différence entre concept individuel et concept générique, se reporter au chapitre 3.

#### 3.1.1 Les noeuds.

Structure d'un noeud : (NumNd NomNd)

où	NumNd	entier (identifiant)
	NomNd	string

Structure de la liste des noeuds : LN=((NumNd NomNd)\*)

#### 3.1.2 Les ensembles.

Structure d'un ensemble :

(NumEns NomEns Quant T\_REL (NumNd\*))

où	NumEns	entier (identifiant)
	NomEns	string
	Quant	(DE, IN, CA)
	(NumNd*)	liste d'entiers (identifiants)
	T_REL	(COLLT, DISJT, DISTR, RESPT)

Structure de la liste des ensembles :

LE=((NumEns NomEns Quant T\_REL (NumNd\*))\*)



### 3.1.3 Les relations entre deux ensembles.

Structure d'une relation : (T\_RELENS NumEns1 NumEns2)

où T\_RELENS (EGAL, INCLUS, INCLUSDIST) +  
RELATIONS DE TYPE CASUEL  
NumEns1 entier (identifiant)  
NumEns2 entier (identifiant)

Structure de la liste des relations :

LRLE = ((T\_RELENS NumEns1 NumEns2)\*)

### 3.1.4 Les références.

Structure d'une référence : (NumRef NumEns NumCg)

où NumRef entier (identifiant)  
NumEns entier (identifiant)  
NumCg entier (identifiant)

Structure de la liste des références :

LRF = ((NumRef NumEns NumCg)\*)

## 3.2 Spécifications internes.

### 3.2.1 Les noeuds.

#### A. Constructeurs.

##### 1° Construire un noeud.

*Syntaxe :* **CONSTRND** (NomNd)

*But :* Construire une paire dont le premier élément est le numéro du noeud et le deuxième le nom du noeud.

*Pré :* /

*Post :* Un noeud de numéro NumNd et de nom NomNd est créé.

*IN :* NumNd                      string

*OUT :* (NumNd NomNd)

où    NumNd                      entier (identifiant)  
      NomNd                      string

##### 2° Ajouter un noeud à une liste de noeuds.

*Syntaxe :* **AJOUTND** ((NumNd NomNd) LN)

*But :* Ajouter à une liste LN une paire dont le premier élément est le numéro du noeud et le deuxième le nom du noeud.

*Pré :* (NumNd NomNd) et LN existent.

*Post :*  $LN = LN \cup (NumNd \text{ NomNd})$ .

*IN :* (NumNd NomNd)    noeud  
      IN                      liste de noeuds

*OUT :* IN                      liste de noeuds

### 3° Enlever un noeud d'une liste de noeuds.

*Syntaxe :* **ENLEVND** ( NumNd LN )

*But :* Enlever d'une liste LN une paire dont le premier élément est le numéro du noeud donné en entrée.

*Pré :* NumNd et LN existent.

*Post :*  $LN = LN - \{(NumNd\ NomNd)\}$ .

*IN :* NumNd entier (identifiant)  
IN liste de noeuds

*OUT :* IN liste de noeuds

### 4° Modifier un noeud.

*Syntaxe :* **MODIFND** ( NumNd NvNomNd )

*But :* Modification d'une paire dont le premier élément numéro du noeud reste inchangé, le deuxième étant remplacé par le nouveau nom du noeud donné en entrée.

*Pré :* le numéro de noeud NumNd existe.

*Post :* le Nom NomNd du noeud numéro NumNd est remplacé par NvNomNd.

*IN :* NumNd entier (identifiant)  
NvNomNd string

*OUT :* NumNd entier (identifiant)  
NvNomNd string

## B. Fonctions d'accès.

### 1° Rechercher un noeud dans une liste selon un critère.

*Syntaxe :* **RECHND**(critère LN)

où le critère peut être un numéro de noeud NumNd.  
un nom de noeud NomNd.

*But :* Rechercher un numéro ou un nom de noeud dans une liste de noeuds.

*Pré :* LN existe.

*Post :* LN est inchangée.

*IN :* NumNd entier (identifiant)  
ou  
NomNd string

IN liste de noeuds : ((NumNd NumNd) \*)

*OUT* • (NumNd NomNd) si le Numéro du noeud NumNd  
ou le Nom du noeud NomNd se trouve dans la liste LN.  
• "Nil" sinon

*Remarque :* Les noeuds ont un Nom NomNd si et seulement si ils sont explicitement instanciés. S'ils ne sont pas explicitement instanciés, le nom est remplacé par le symbole "\*".

Exemples : "Jean" → (10 Jean)  
"Nancy" → (23 Nancy)  
\* → (13 Nil)

### 3.2.2 Les ensembles.

#### A. Constructeurs.

##### 1° Construire un ensemble.

*Syntaxe :* **CONSTRENS**(NomEns Quant REL (NomNd\*))

*But :* Construire un quintuplet composé comme suit : le premier élément est le numéro de l'ensemble, le deuxième le nom de cet ensemble, le troisième son type de quantificateur, le quatrième la relation existante entre les éléments de cet ensemble et le cinquième la liste des numéros des noeuds qui lui appartiennent.

*Pré :* /

*Post :* Un ensemble de numéro NumEns, de type de quantificateur Quant, de relation REL et de nom NomEns est créé. Ses éléments sont (NumNd\*).

<i>IN :</i>	NumEns	string
	Quant	(DE, IN, CA)
	REL	(COLLT, DISJT, DISTR, RESPT)
	(NomNd*)	liste de string

*OUT :* (NumEns NomEns Quant REL (NumNd\*))

où	NumEns	entier (identifiant)
	NomEns	string
	Quant	(DE, IN, CA)
	REL	(COLLT, DISJT, DISTR, RESPT)
	(NumNd*)	liste de numéro de noeuds.

##### 2° Ajouter un ensemble à une liste d'ensemble.

*Syntaxe :* **AJOUTENS**((NumEns NomEns Quant REL (NumNd\*)) LE)

*But :* Ajouter à une liste un quintuplet composé comme suit : le premier élément est le numéro de l'ensemble, le deuxième le nom de cet ensemble, le troisième son type de quantificateur, le quatrième est la relation existante entre les éléments de cet ensemble et le

cinquième la liste des numéros des noeuds qui lui appartiennent.

*Pré :* (NumEns NomEns Quant REL (NumNd\*)) et LE existent.

*Post :*  $LE = LE \cup \{(NumEns\ NomEns\ Quant\ REL\ (NumNd*))\}$ .

*IN :* (NumEns NomEns Quant REL (NumNd\*)) un ensemble  
IE une liste  
d'ensembles

*OUT :* IE une liste d'ensembles.

### 3° Enlever un ensemble d'une liste d'ensemble.

*Syntaxe :* **ENLEVENS** (NumEns LE)

*But :* Enlever d'une liste un quintuplet dont le premier élément est le numéro de l'ensemble donné en entrée.

*Pré :* NumEns et LE existent.

*Post :*  $LE = LE - \{(NumEns\ NomEns\ Quant\ REL\ (NumNd*))\}$ .

*IN :* NumEns entier (identifiant)  
IE une liste d'ensembles

*OUT :* IE une liste d'ensembles.

### 4° Construire une référence.

*Syntaxe :* **CONSTREF** (NumEns NumCG)

*But :* construire un triplet dont le premier élément est un numéro de référence, le deuxième un numéro d'ensemble et le troisième un numéro de concept générique.

*Pré :* Le numéro de l'ensemble NumEns et le numéro du concept générique NumCG existent.

*Post :* Une référence est créée.

*IN :* NumEns entier (identifiant)  
 NumCG entier (identifiant)

*OUT :* (NumRef NumEns NomEns)  
 où NumRef entier (identifiant)

#### 5° Ajouter une référence à une liste de références.

*Syntaxe :* **AJOUTREF**((NumRef NumEns NumCg) LRF)

*But :* Ajouter à une liste un triplet composé comme suit : le premier élément est le numéro de la référence, le deuxième le numéro de l'ensemble et le troisième le numéro du concept générique.

*Pré :* (NumRef NumEns NumCg) et LRF existent.

*Post :*  $LRF = LRF \cup \{(NumRef\ NumEns\ NumCg)\}$ .

*IN :* (NumRef NumEns NumCg) une référence  
 LRF une liste de références.

*OUT :* LRF une liste de références.

#### 6° Enlever une référence d'une liste de références.

*Syntaxe :* **ENLEVREF**(NumRef LRF)

*But :* Enlever d'une liste un triplet dont le premier élément est le numéro de la référence.

*Pré :* NumRef et LRF existent.

*Post :*  $LRF = LRF - \{(NumRef\ NumEns\ NumCg)\}$ .

*IN :* NumRef un numéro de référence  
 LRF une liste de références

*OUT :* LRF une liste de références.

7° Construire une relation entre deux ensembles.

*Syntaxe* : **CONSTRELENS**(T\_RELENS NumEns1 NumEns2)

*But* : Construire un triplet composé comme suit : le premier élément est la relation, le deuxième et le troisième sont les numéros des ensembles qui participent à cette relation.

*Pré* : NumEns1 et NumEns2 existent.

*Post* : Une relation entre deux ensembles est créée.

*IN* : T\_RELENS (EGAL, INCLUS, INCLUSDIST) + Relations de type casuel  
NumEns1 entier (identifiant)  
NumEns2 entier (identifiant)

*OUT* : (T\_RELENS NumEns1 NumEns2) une relation

8° Ajouter une relation à une liste de relations.

*Syntaxe* : **AJOUTRELENS**((T\_RELENS NumEns1 NumEns2) LRLE)

*But* : Ajouter à une liste de relations un triplet composé comme suit : le premier élément est la relation, le deuxième et le troisième sont les numéros des ensembles qui participent à cette relation.

*Pré* : (T\_RELENS NumEns1 NumEns2) et LRLE existent.

*Post* :  $LRLE = LRLE \cup \{(T\_RELENS \text{ NumEns1 NumEns2})\}$ .

*IN* : (T\_RELENS NumEns1 NumEns2) une relation  
LRLE une liste de relations

*OUT* : LRLE une liste de relations

9° Enlever une relation à une liste de relations.

*Syntaxe* : **ENLEVRELENS**((T\_RELENS NumEns1 NumEns2) LRLE)

*But* : Enlever d'une liste de relations un triplet composé comme suit : le premier élément est la relation, le



deuxième et le troisième sont les numéros des ensembles qui participent à cette relation.

*Pré :* (T\_RELENS NumEns1 NumEns2) et LRLE existent.

*Post :* LRLE = LRLE - {(T\_RELENS NumEns1 NumEns2)}.

*IN :* (T\_RELENS NumEns1 NumEns2) une relation  
LRLE une liste de relations

*OUT :* LRLE une liste de relations

#### 10° Modifier une relation.

*Syntaxe :* **MODIFREL** ( NumEns1 NumEns2 NVT\_REL)

*But :* Modifier une relation entre deux ensembles.

*Pré :* Il existe une relation T\_REL entre Les deux ensembles de numéros NumEns1 et NumEns2 (Qui existent également).

*Post :* une nouvelle relation NVT\_REL entre les numéros des ensembles NumEns1 et NumEns2 est créée.

*IN :* NVT\_REL (EGAL, INCLUS, INCLUSDIST)  
+ RELATIONS DE TYPE CASUEL.  
NumEns1 entier (identifiant)  
NumEns2 entier (identifiant)

*OUT :* (NVT\_REL NumEns1 NumEns2)

#### 11° Modifier un ensemble.

*Syntaxe :* **MODIFENS**( NumEns critère)

*But :* Modifier un ensemble dont le numéro est donné en entrée. Le critère indique la partie remplaceante de l'ensemble. Ce critère peut être :

- 1° le nom de l'ensemble.
- 2° le type de quantificateur.
- 3° la relation entre les éléments de cet ensemble.
- 4° la liste des numéros des éléments de cet ensemble.

*Pré :* Le numéro de l'ensemble NumEns existent ainsi que le critère.

*Post :* L'ensemble de numéro NumEns est modifié selon le critère.

*IN :* NumEns entier (identifiant)  
critère= NvNomEns string  
NvQuant (DE, IN, CA)  
NvT\_REL (COLLT, DISJT, DISTR,  
RESPT)  
(NvNumNd\*) liste d'entiers identifiants

*OUT :* (NumEns NomEns Quant REL (NumNd\*)) où le critère a remplacé son élément correspondant.

## 12° Modifier une référence.

*Syntaxe :* **MODIFREF** ( NumRef NvNumEns1 NvNumEns2 )

*But :* Modifier un triplet dont le premier élément est le numéro de référence donné en entrée, les deux derniers éléments donnés en entrée étant les nouveaux numéros de ensemble participant à la référence de numéro donné.

*Pré :* NumRef, NvNumEns1 et NvNumEns2 existent.

*Post :* La référence de numéro NumRef est modifiée.

*IN :* NumRef entier (identifiant)  
NvNumEns1 entier (identifiant)  
NvNumEns2 entier (identifiant)

*OUT :* ( NumRef NvNumEns1 NvNumEns2 )

## B. Fonctions d'accès.

### 1° Rechercher un ensemble selon un critère.

*Syntaxe :* **RECHENS**(critère LE)

*But :* Rechercher dans une liste un ensemble selon un certain critère. Ce critère peut être :

- 1° le numéro de l'ensemble.
- 2° le nom de l'ensemble.
- 3° la liste des numéros des éléments de cet ensemble.
- 4° la liste des noms des éléments de cet ensemble.

*Pré :* LE existe.

*Post :* LE est inchangé.

*IN :*

IE	liste d'ensembles.
critère=	NomEns string
	NumEns entier (identifiant)
	(NumNd*) liste d'entiers identifiants
	(NomNd*) liste de string

*OUT :*

- (NumEns NomEns Quant T\_REL (NumNd\*)) si le critère est trouvé dans LE.
- "Nil" sinon.

### 2° Rechercher un type de relation.

*Syntaxe :* **RECHRELENS**(NumEns1 NumEns2 LRLE)

*But :* Rechercher dans une liste de relations une relation à laquelle participent les deux numéros d'ensembles donnés en entrée.

*Pré :* NumEns1, NumEns2 et LRLE existent.

*Post :* LRLE est inchangée.

$IN :$	NumEns1	entier (identifiant)
	NumEns2	entier (identifiant)
	IRLE	liste de relations

*OUT:*

- (T\_RELENS NumEns1 NumEns2) s'il existe une relation T\_RELENS entre les ensembles de numéros NumEns1 et NumEns2.

où	T_RELENS	(EGAL, INCLUS, INCLUSDIST) + RELATIONS DE TYPE CASUEL
	NumEns1	entier (identifiant)
	NumEns2	entier (identifiant)

- "Nil" sinon.

3° Rechercher une référence.

**Syntax :** RECHREF ( NumEns NumCq LRF)

**But :** Rechercher dans une liste de références une référence ayant comme numéro d'ensemble et de concept générique ceux donnés en entrée.

*Pré :* NumEns, NumCq et LRF existent.

*Post :* LRF est inchangé.

<i>IN</i> :	NumEns	entier (identifiant)
	NumCg	entier (identifiant)
	IRF	liste de références.

*OUT*: • (NumRef NumEns NumCg) s'il existe une référence entre NumEns et NumCg dans la liste LRF.

- "Nil" sinon.

### **3.2.3. Les formules.**

#### **A. Constructeurs.**

1° Construire une formule.

2° Détruire une formule.

3° Modifier une formule.

## **Annexe B**

### **Listing des programmes**

## Les constructeurs

```
(in-package "USER")
```

```
(defun creerlistecp (listetemp)
  (when listetemp
    (progn
      (setq compteur (+ 1 compteur))
      (cons (list compteur (car listetemp)) (creerlistecp (rest
listetemp))))
  )
)
)
```

---

```
(defun ajoutcp (cp l)
  (if (not (rechcp cp l))
    (progn (setq compteur (+ 1 compteur))
      (append l (list (list compteur cp))))
    (values l)
  )
)
```

---

```
(defun creerlisterel (listetempo y)
  (when listetempo
    (cons (list (car (car listetempo)) (rechcp (second (car listetempo))
y)
      (rechcp (third (car listetempo)) y))
      (creerlisterel (rest listetempo) y))
  )
)
```

---

```
(defun creerlistepr (l)
  (when l
    (cons (list (rechcp (second (car l)) LCG) (first (car l)))
      (creerlistepr (rest l)))
  ))
```

---

```

(defun ajoutpr (pr cp l)
  (let ((num (rechcp cp LCG)))
    (if (not (member (list num pr) l :test #'equal))
        (append l (list (list num pr)))
        (values l))
    )
  )
)

```

---

```

(defun constci (l)
  (when l
    (let ((tempo (rechcp (car l) LN)))
      (if tempo
          (cons tempo (constci (cdr l)))
          (let ((res (creerlistecp (list (car l)))))
              (setq LN (append LN res))
              (cons (car (car res)) (constci (cdr l))))
          )
      )
  )
)

```

---

```

(defun constlistedeic (l)
  (when l
    (setq compteur (+ 1 compteur))
    (cons (list compteur (first (car l))) (constlistedeic (rest l)))
  )
)

```

---

```

(defun constrlisteensdeic (l)
  (when l
    (cons (consens '(n=1) 'de '()) (list(first (car l))))
    (constrlisteensdeic (rest l)))
  )
)

```

---

```

(defun constrel (rel numens1 numens2)
  (list rel (rechcp numens1 LE) (rechcp numens2 LE))
)

```

---



```

(defun ajoutrel (rel numens1 numens2 l)
  (if (not (member (list rel numens1 numens2) l :test #'equal))
      (append l (list (list rel numens1 numens2))))
      (values l)
  )
)

```

---

```

(defun constref (numens numcg)
  (setq compteur (+ 1 compteur))
  (list compteur (rechcp numens LE) (rechcp numcg LCG))
)

```

---

```

(defun creerlisteref (l ld)
  (when l
    (cons (constref (first (car l)) (second (car ld)))
          (creerlisteref (rest l) (rest ld))))
  )
)

```

---

```

(defun ajoutref (numens numcg l)
  (if (not (rechrel numens numcg l))
      (append l (list (constref numens numcg))))
      (values l)
  )
)

```

---

```

(defun consens (cond quant rel l)
  (setq compteur (+ 1 compteur))
  (list compteur cond quant rel (constci l)))

```

```

(defun ajoutens (cond quant rel l lens)
  (append lens (list (consens cond quant rel l))))
)

```

---

```

(defun constrlistenum (l lrech)
  (when l
    (cons (rechcp (car l) lrech) (constrlistenum (rest l) lrech))
  )
)

```

---

```

(defun enleve (num l)
  (when l
    (if (equal (first (car l)) num)
        (enleve num (rest l))
        (cons (car l) (enleve num (rest l))))
  )
)

```

---

```

(defun enleverel (numens1 numens2 l)
  (when l
    (if (or (and (equal (second (car l)) numens1)
                  (equal (third (car l)) numens2))
            (and (equal (third (car l)) numens1)
                  (equal (second (car l)) numens2)))
        (enleverel numens1 numens2 (rest l))
        (cons (car l) (enleverel numens1 numens2 (rest l))))
  )
)

```

---

```

(defun modifcp (num nvnom l)
  (when l
    (if (equal (first (car l)) num)
        (cons (list num nvnom)
              (modifcp num nvnom (rest l)))
        (cons (car l) (modifcp num nvnom (rest l))))
  )
)

```

---

```

(defun modifrel (numens1 numens2 nvrel l)
  (when l
    (if (and (equal (second (car l)) numens1)
              (equal (third (car l)) numens2))
        (cons (list nvrel numens1 numens2)
              (modifrel numens1 numens2 nvrel (rest l)))
        (cons (car l) (modifrel numens1 numens2 nvrel (rest l))))
    )
  )
)

```

---

```

(defun modifens (numens critere l)
  (when l
    (if (equal (first (car l)) numens)
        (cond ((equal (type-of critere) 'symbol)
              (cond ((member critere '(de in ca))
                    (cons (list numens (second (car l)) critere
                              (fourth (car l)) (fifth (car l)))
                          (modifens numens critere (rest l)))))
              ((member critere '(collt disjt disct respt distr))
               (cons (list numens (second (car l))
                              (third (car l)) critere
                              (fifth (car l)))
                     (modifens numens critere (rest l)))))
              ((equal (type-of (first critere)) 'fixnum)
               (cons (list numens (second (car l))
                              (third (car l)) (fourth (car l))
                              critere)
                     (modifens numens critere (rest l)))))
              ((equal (type-of (first critere)) 'symbol)
               (cons (list numens critere
                              (third (car l)) (fourth (car l))
                              (fifth (car l)))
                     (modifens numens critere (rest l)))))
        )
      (cons (car l) (modifens numens critere (rest l)))
    )
  )
)

```

---

```

(defun modifref (numref nvnumens nvnumcg l)
  (when l
    (if (equal (first (car l)) numref)
        (cons (list numref (rechcp nvnumens LE) (rechcp nvnumcg
LCG))
              (modifref numref nvnumens nvnumcg (rest l)))
        (cons (car l) (modifref numref nvnumens nvnumcg (rest l)))
    )
  )
)

```

## Les fonctions d'accès

```
(in-package "USER")
```

```
(defun rechcp (critere l)
  (when l
    (if (equal (type-of critere) 'symbol)
        (if (equal critere (second (car l)))
            (values (first (car l)))
            (rechcp critere (rest l)))
        (if (equal critere (first (car l)))
            (values (first (car l)))
            (rechcp critere (rest l))))))
  )
)
```

---

```
(defun rechrelpere (rel num l)
  (when l
    (if (and (equal rel (first (car l))) (equal num (second (car l))))
        (cons (car l) (rechrelpere rel num (rest l)))
        (rechrelpere rel num (rest l))))
  )
)
```

---

```
(defun rechrelfils (rel num l)
  (when l
    (if (and (equal rel (first (car l))) (equal num (third (car l))))
        (cons (car l) (rechrelfils rel num (rest l)))
        (rechrelfils rel num (rest l))))
  )
)
```

---

```
(defun rechrel (num1 num2 l)
  (when l
```

```

        (if (or (and (equal num1 (second (car l))) (equal num2 (third
(car l))))
            (and (equal num1 (third (car l))) (equal num2 (second (car
l)))))
        (cons (car l) (rechrel num1 num2 (rest l)))
        (rechrel num1 num2 (rest l)))
    )
)
-----

```

```

(defun rechlisterelempere (num l)
  (when l
    (if (equal (second (car l)) num)
        (cons (car l) (rechlisterelempere num (rest l)))
        (rechlisterelempere num (rest l)))
    )
  )
)
-----

```

```

(defun rechlisterelempils (num l)
  (when l
    (if (equal (third (car l)) num)
        (cons (car l) (rechlisterelempils num (rest l)))
        (rechlisterelempils num (rest l)))
    )
  )
)
-----

```

```

(defun rechnum (num l)
  (when l
    (if (equal (first (car l)) num)
        (values (car l))
        (rechnum num (rest l)))
    )
  )
)
-----

```

```

(defun rechlistenum (lnum l)
  (when l
    (if (equal (fifth (car l)) lnum)
        (values (car l))
        (rechlistenum lnum (rest l)))
    )))
-----

```

```

(defun rechens (critere)
  (if (equal (type-of critere) 'fixnum)
      (rechnum critere LE)
      (if (equal (type-of (car critere)) 'fixnum)
          (rechlistenum critere LE)
          (rechlistenum (constrlistenum critere LN) LE)
      )
  )
)

```

---

```

(defun rechrefens (ens l)
  (when l
    (if (equal (second (car l)) ens)
        (values (car l))
        (rechrefens ens (rest l))
    )
  )
)

```

---

```

(defun rechrefcg (cg l)
  (when l
    (if (equal (third (car l)) (rechcp cg LCG))
        (cons (car l) (rechrefcg cg (rest l)))
        (rechrefcg cg (rest l))
    )
  )
)

```

## Les fonctions d'entrée-sortie

```
(in-package "USER")
```

```
(defun lire (l)
  (when l
    (progn
      (print (car l))
      (lire (rest l))
    )
  )
)
```

---

```
(defun lectfich (nom_fichier)
  (with-open-file (listentiere nom_fichier :direction :input)
    (read listentiere nil)))
```

---

```
(defun ecriture (l nom_fichier)
  (with-open-file (temp1 nom_fichier :direction :output)
    (dolist (elem l nil)
      (print elem temp1))))
```

---

```
(defun affichnom (l)
  (when l
    (cons (second (rechner (first l) LN)) (affichnom (rest l)))
  )
)
```

---

```
(defun affiche (laf)
  (setq i (+ 1 i))
  (when laf
    (let ((temp (car laf)))
      (format t "Ens n ~d : [~S ~S ~S ~S ~S]~%" i (first temp)
              (second temp) (third temp)
              (fourth temp) (affichnom (fifth temp)))
    )
    (affiche (rest laf))
  ))
```

---



```

(defun afficherel (l)
  (setq j (+ 1 j))
  (when l
    (let ((temp0 (first (car l)))
          (temp1 (rechner (second (car l)) LE))
          (temp2 (rechner (third (car l)) LE)))
      (format t "Rel n ~d : "j)
      (afficheens temp1)
      (format t " => (~S) => "temp0)
      (afficheens temp2)
      (format t "~%"))
    (afficherel (rest l)))
  )
)

```

---

```

(defun afficheens (ens)
  (format t "[~S ~S ~S ~S ~S]"(first ens)
          (second ens) (third ens)
          (fourth ens) (afficheens (fifth ens)))
  )
)

```

---

```

(defun afficheref (l)
  (when l
    (format t "~S : "(first (car l)))
    (afficheens (rechner (second (car l)) LE))
    (format t " : ~S~%" (second (rechner (third (car l)) LCG)))
    (afficheref (rest l)))
  )
)

```

## **L'initialisation générale**

```
(in-package "USER")

(load "initvarglobales")
(load "in-out.lisp")
(load "fonctions_d_acces")
(load "constructeurs")
(load "initgraphconcptl")
(load "initprimsup")
(load "inithistodial")
```

## **L'initialisation des variables globales**

```
(in-package "USER")

(defvar compteur 0)
(defvar LPR ())
(defvar LCG ())
(defvar LRCH ())
(defvar LINST ())
(defvar LN ())
(defvar LE ())
(defvar LRLE ())
(defvar LRF ())
```

## **L'initialisation du graphe conceptuel**

```
(in-package "USER")

(setq intcg (lectfich "~druart/=lisp/=donnees/fcg.data"))
(setq LCG (creerlistecp intcg))
(ecriture LCG "~druart/=lisp/=resultats/rescg.data")

(setq intrel (lectfich "~druart/=lisp/=donnees/frelcg.data"))
(setq LRCH (creerlisterel intrel LCG))
(ecriture LRCH "~druart/=lisp/=resultats/resrelcg.data")
```

## L'initialisation des primitives supérieures

```
(in-package "USER")
```

```
(setq intpr (lectfich "~druart/=lisp/=donnees/fprimsup.data"))  
(setq LPR (creerlistepr intpr))  
(ecriture LPR "~druart/=lisp/=resultats/resprimsup.data")
```

## L'initialisation de l'historique du dialogue

```
(in-package "USER")
```

```
(setq intind (lectfich "~druart/=lisp/=donnees/fdeictic.data"))  
(setq LN (constlistedeic intind))  
(ecriture LN "~druart/=lisp/=resultats/resdeictic.data")
```

```
(setq LE (constrlisteensdeic intind))  
(ecriture LE "~druart/=lisp/=resultats/resens.data")  
(setq LRF (creerlisteref LE intind))  
(ecriture LRF "~druart/=lisp/=resultats/resref.data")
```

## Le programme de démonstration

```
(in-package "USER")
```

```
(load "initgenerale")  
(defun creens (cond quant rel lnom)  
  (let ((tempens (consens cond quant rel lnom)))  
    (setq LE (append LE (list tempens)))  
    (setq i 0)  
    (format t "~%~%")  
    (format t "Voici la liste des ensembles :~%")  
    (format t "-----~%")  
    (format t "~%")  
    (affiche LE)  
    (majref (car tempens) lnom)  
    (afficheref LRF)  
  )  
)
```

---

```

(defun majref (numens lnom)
  (let ((temprem (remove-duplicates (constrlistenum lnom LCG))))

    (if (equal (length temprem) 1)
      (setq LRF (ajoutref numens (first temprem) LRF))
      (progn
        (rechcgcommun temprem)
        (setq LRF (ajoutref numens r LRF))))
    )
    (format t "~%~%La reference a ete creee~%~%")
    (format t "Ref :   Ensemble           :           Concept generique~%")
    (format t "-----
~%~%")
  )
)

```

```

-----

(defun rechcgcommun (listenum)
  (when (not (equal (length listenum) 1))
    (if (equal (comparelem (first listenum) (second listenum))) 'ok)
    (rechcgcommun (cons (first listenum) (rest (rest
listenum))))))
    (rechcgcommun
      (cons (third (car (rechrelpere 'sorte_de (first listenum)
LRCH)))
        (rest listenum)))
    )
  )
  (if (equal (length listenum) 1)
    (setq r (first listenum)))
  )
)

```

```

-----

(defun comparelem (elem1 elem2)
  (if (equal elem1 elem2)
    (values 'ok)
    (if (and (not (equal (third (car (rechrelpere 'sorte_de elem2
LRCH)))
      'nil))
      (not (equal elem1 'nil)))
      (if (equal (third (car (rechrelpere 'sorte_de elem2 LRCH)))
elem1)
        (progn (comparelem 'nil 'nil)
          (values 'ok))
        )
      )
    )
  )
)

```

```

      (comparelem elem1 (third (car (rechrelpere 'sorte_de elem2
LRCH))))))
    )
  ))

```

---

```

(defun crerelens (rel num1 num2)
  (setq LRLE (ajoutrel rel num1 num2 lrle))
  (setq j 0)
  (format t "~%")
  (format t "Voici la liste des relations entre les ensembles :~%")
  (format t "-----
~%~%~%")
  (afficherel LRLE)
  (format t "~%~%~%")
)

```

---

```

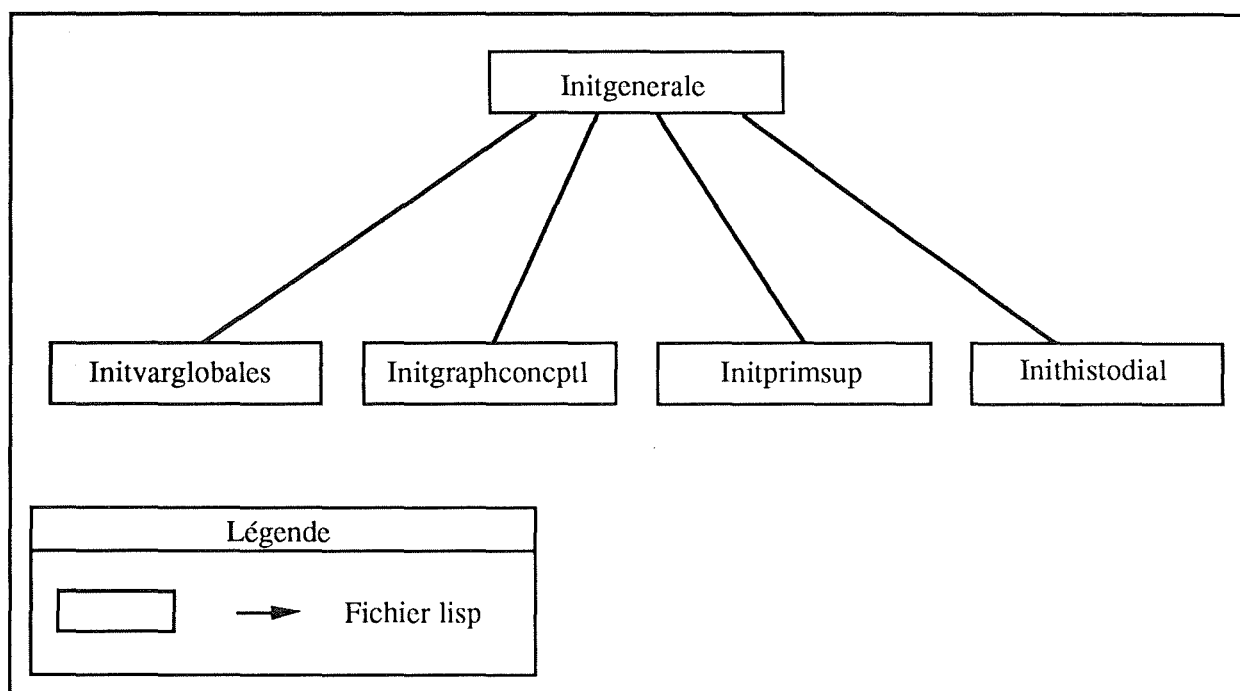
(defun creref (numens numcg)
  (setq LRF (ajoutref numens numcg LRF))
  (format t "~%~%")
  (format t "Ref : Ensemble : Concept generique~%")
  (format t "-----
~%~%")
  (afficheref LRF)
)

```

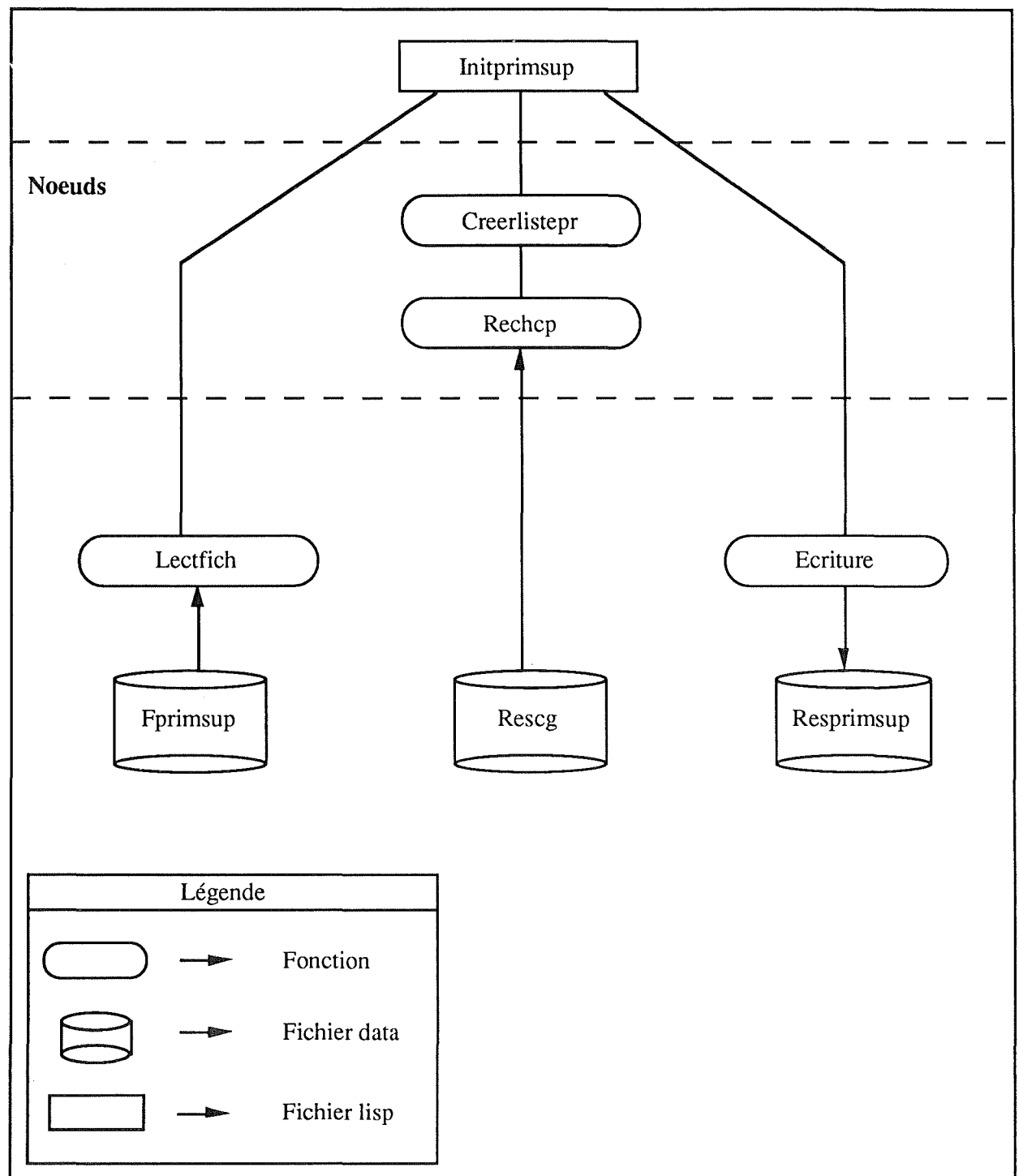
## **Annexe C**

### **Organigramme général de l'application**

## L'initialisation générale

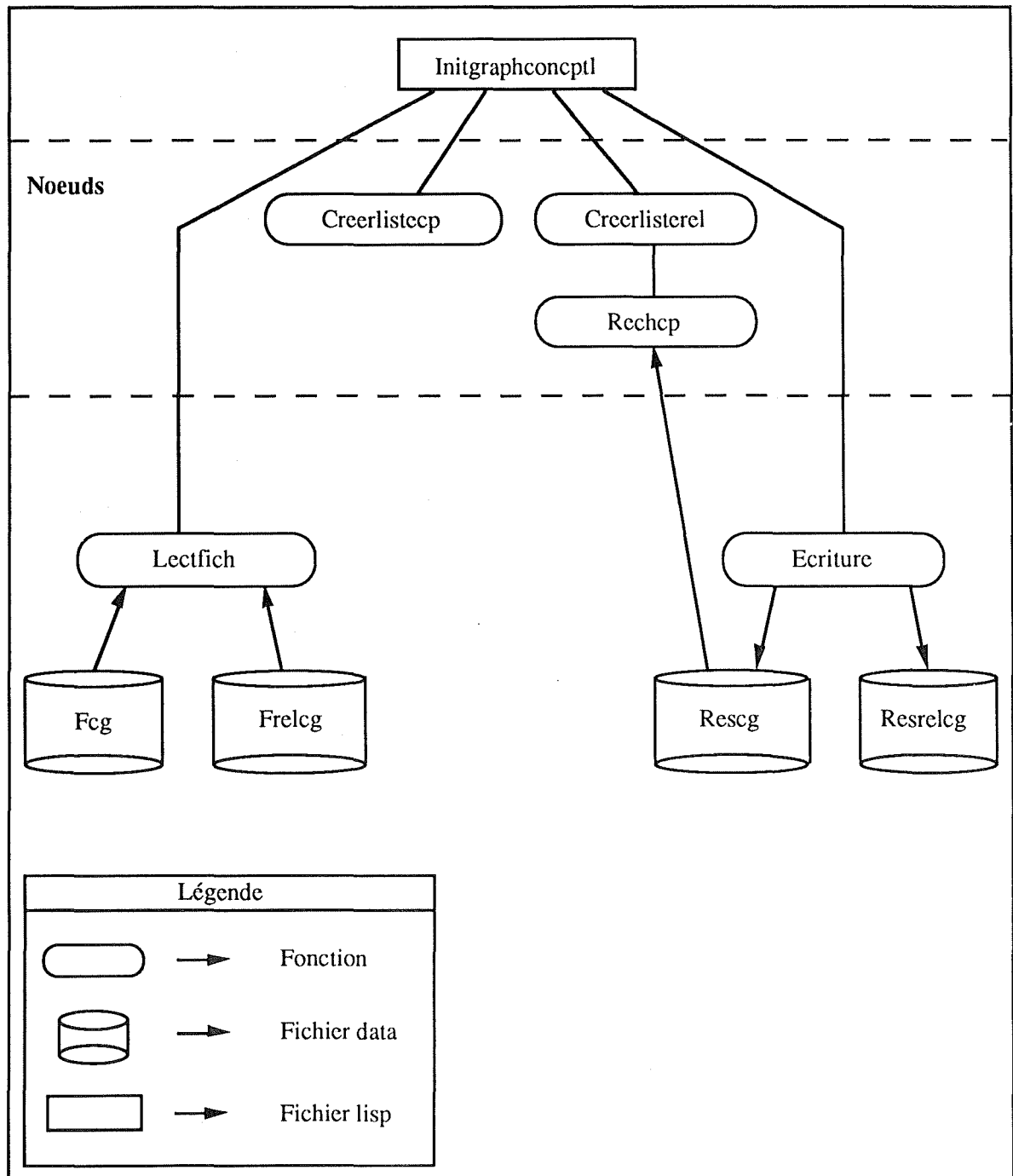


## L'initialisation de la couche supérieure

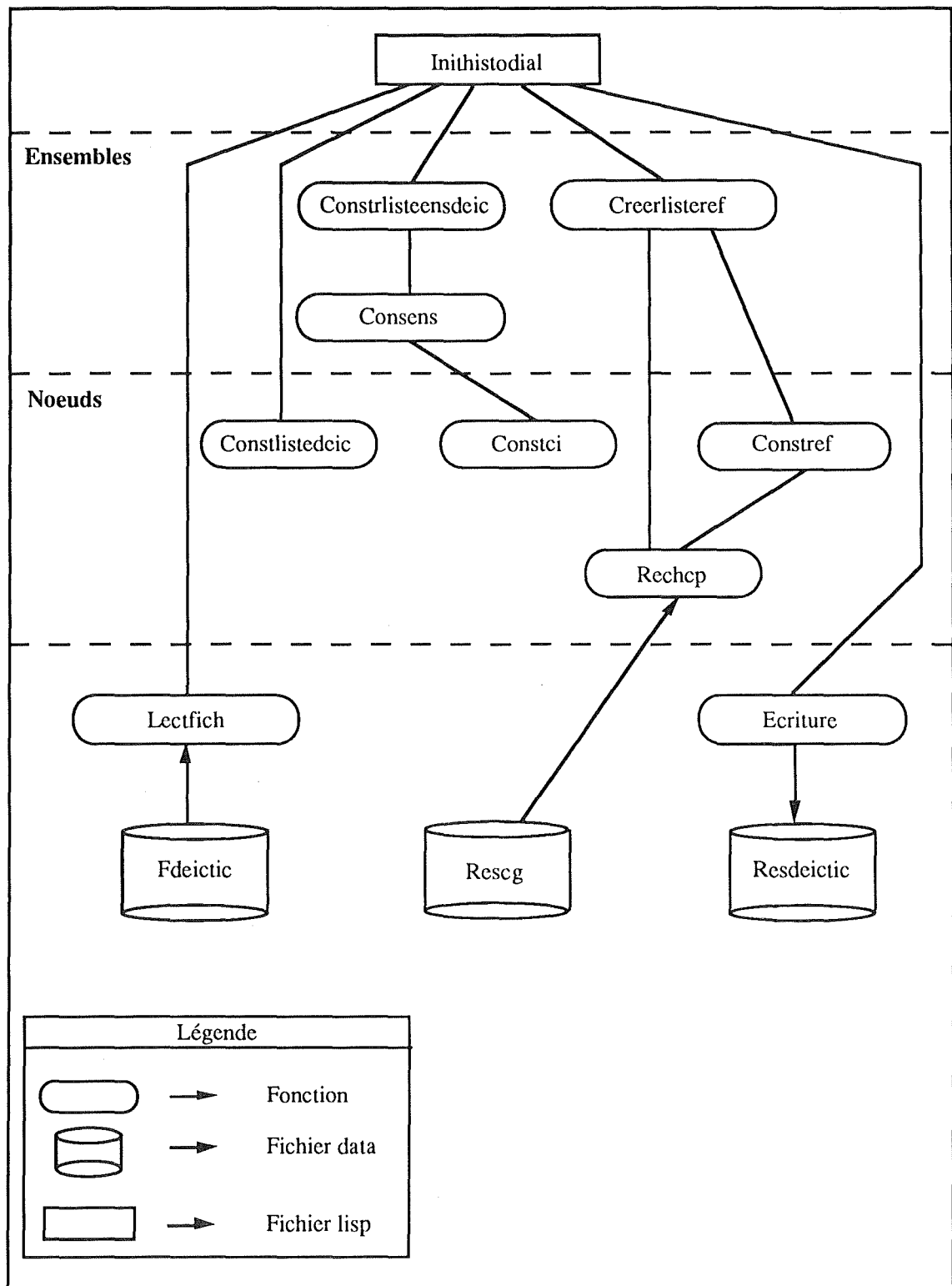




## L'initialisation du graphe conceptuel



## L'initialisation de l'historique du dialogue



## **Annexe D**

### **Données**

## Les primitives supérieures

((mvmt\_2 aller) (act\_2 renouveler) (act\_2 modifier) (act\_2 etablr)  
(cognt\_3 autoriser) (act\_2 annuler) (echobt\_2 obtenir) (cognt\_3  
declarer) (act\_2 remplir) (act\_2 munir) (echprod\_3 fournir) (act\_2  
presenter) (act\_2 accompagner) (act\_2 joindre) (cognt\_1 ecouter)  
(cognt\_2 ecouter) (echprod\_3 donner) (cognt\_3 demander) (act\_2  
ecrire) (cognt\_3 appeler) (cognt\_3 prouver) (statut\_2  
etre\_departement))

## Les concepts génériques

(inanime  
  lieu  
    ville  
      nancy  
      vandoeuvre  
      villers  
      laxou  
      maxeville  
      malzeville  
      paris  
      metz  
    departement  
      moselle  
      meurthe\_et\_moselle  
      meuse  
      vosges  
      region\_parisienne  
  commerce  
    bureau\_de\_tabac  
    supermarche  
  administration  
    sp\_p  
      prefecture  
      sous\_prefecture  
    mairie  
    hotel\_de\_ville  
    commissariat  
    gendarmerie  
  cira  
  cpam

- asina
  - ass\_consommateurs
  - ophlm
  - consulat
    - consulat\_d\_algerie
    - consulat\_de\_rfa
    - consulat\_du\_maroc
- pays
  - france
  - etranger
    - cee
      - belgique
      - luxembourg
      - rfa
      - italie
      - pays\_bas
      - espagne
      - portugal
      - grece
      - danemark
      - grande\_bretagne
        - angleterre
        - ecosse
        - irlande
        - pays\_de\_galles
    - non\_cee
      - usa
      - urss
      - madagascar
      - maroc
      - algerie
- document
  - piece\_administrative
    - carte\_d'identite
    - passeport
    - carte\_grise
    - fiche\_d'etat\_civil
    - livret\_de\_famille
    - carte\_de\_sejour
    - certificat\_de\_scolarite
    - autorisation\_de\_sortie
    - declaration\_de\_vol
    - declaration\_de\_perte
    - visa
    - carte\_ss
  - certificat\_medical

lettre  
formulaire  
photo  
timbre  
taxe  
impots  
tva  
impots\_locaux  
evenement  
deces  
mariage  
naissance  
perte  
vol  
anime  
personne  
locuteur  
machine  
tiers  
animal  
domestique  
chien  
chat  
poisson  
sauvage  
unite  
unite\_prix  
franc  
centime  
unite\_taille  
metre  
centimetre  
unite\_temps  
an  
mois  
jour  
heure  
coordonnee  
residence  
telephone  
adresse  
identite  
nom  
nom\_jf  
religion

- chretien
  - musulman
  - juif
  - athee
  - hindou
  - bouddhiste
- etat
  - couleur
    - bleu
    - rouge
    - jaune
    - rose
    - gris
    - noir
    - blanc
  - qualificatif
    - ancien
    - nouveau
    - grand
    - moyen
    - petit
    - prorogable
    - renouvelable
    - perime
    - valide
  - etat\_legal
    - responsabilite
      - majeur
      - mineur
  - nationalite
  - situation
    - marie
    - divorce
    - veuf
    - celibataire
    - remarie
  - sexe
    - masculin
    - feminin
  - changement
- aller
- renouveler
- modifier
- etablir
- autoriser
- annuler

obtenir  
declarer  
remplir  
munir  
fournir  
presenter  
accompagner  
joindre  
ecouter  
donner  
demander  
ecrire  
rencontrer  
appeler  
prouver  
etre\_departement

)

## Les relations entre les concepts génériques

(  
(sorte\_de lieu inanime)  
(sorte\_de ville lieu)  
(sorte\_de nancy ville)  
(sorte\_de vandoeuvre ville)  
(sorte\_de villers ville)  
(sorte\_de laxou ville)  
(sorte\_de malzeville ville)  
(sorte\_de paris ville)  
(sorte\_de metz ville)  
  
(sorte\_de departement lieu)  
(sorte\_de moselle departement)  
(sorte\_de meurthe\_et\_moselle departement)  
(sorte\_de meuse departement)  
(sorte\_de vosges departement)  
(sorte\_de region\_parisienne departement)  
  
(sorte\_de commerce lieu)  
(sorte\_de bureau\_de\_tabac commerce)  
(sorte\_de supermarche commerce)



(sorte\_de administration lieu)  
 (sorte\_de sp\_p administration)  
 (sorte\_de prefecture sp\_p)  
 (sorte\_de sous\_prefecture sp\_p)

(sorte\_de mairie administration)  
 (sorte\_de hotel\_de\_ville administration)  
 (sorte\_de commissariat administration)  
 (sorte\_de gendarmerie administration)  
 (sorte\_de cira administration)  
 (sorte\_de cpam administration)  
 (sorte\_de asina administration)  
 (sorte\_de ass\_consommateurs administration)  
 (sorte\_de ophlm administration)  
 (sorte\_de consulat administration)  
 (sorte\_de consulat\_d\_algerie consulat)  
 (sorte\_de consulat\_de\_rfa consulat)  
 (sorte\_de consulat\_du\_maroc consulat)

(sorte\_de pays lieu)  
 (sorte\_de france pays)

(sorte\_de etranger pays)  
 (sorte\_de cee etranger)  
 (sorte\_de belgique cee)  
 (sorte\_de luxembourg cee)  
 (sorte\_de rfa cee)  
 (sorte\_de italie cee)  
 (sorte\_de pays\_bas cee)  
 (sorte\_de espagne cee)  
 (sorte\_de portugal cee)  
 (sorte\_de grece cee)  
 (sorte\_de danemark cee)  
 (sorte\_de grande\_bretagne cee)  
 (sorte\_de angleterre grande\_bretagne)  
 (sorte\_de ecosse grande\_bretagne)  
 (sorte\_de irlande grande\_bretagne)  
 (sorte\_de pays\_de\_galles grande\_bretagne)

(sorte\_de non\_ee etranger)  
 (sorte\_de usa non\_ee)  
 (sorte\_de urss non\_ee)  
 (sorte\_de madagascar non\_ee)  
 (sorte\_de maroc non\_ee)  
 (sorte\_de algerie non\_ee)

(sorte\_de document inanime)  
(sorte\_de piece\_administrative document)  
(sorte\_de carte\_d\_identite piece\_administrative)  
(sorte\_de passeport piece\_administrative )  
(sorte\_de carte\_grise piece\_administrative)  
(sorte\_de fiche\_d\_etat\_civil piece\_administrative)  
(sorte\_de livret\_de\_famille piece\_administrative)  
(sorte\_de carte\_de\_sejour piece\_administrative)  
(sorte\_de certificat\_de\_scolarite piece\_administrative)  
(sorte\_de autorisation\_de\_sortie piece\_administrative)  
(sorte\_de declaration\_de\_vol piece\_administrative)  
(sorte\_de declaration\_de\_perte piece\_administrative)  
(sorte\_de visa piece\_administrative)  
(sorte\_de carte\_ss piece\_administrative)

(sorte\_de certificat\_medical document)  
(sorte\_de lettre document)  
(sorte\_de photo document)  
(sorte\_de timbre document)

(sorte\_de taxe inanime)  
(sorte\_de impots taxe)  
(sorte\_de tva taxe)  
(sorte\_de impots\_locaux taxe)

(sorte\_de deces evenement)  
(sorte\_de mariage evenement)  
(sorte\_de naissance evenement)  
(sorte\_de perte evenement)  
(sorte\_de vol evenement)

(sorte\_de personne anime)  
(sorte\_de locuteur personne)  
(sorte\_de machine personne)  
(sorte\_de tiers personne)

(sorte\_de animal anime)  
(sorte\_de domestique animal)  
(sorte\_de chien domestique)  
(sorte\_de chat domestique)  
(sorte\_de poisson domestique)  
(sorte\_de sauvage animal)

(sorte\_de administration anime)

(sorte\_de unite\_prix unite)

(sorte\_de franc unite\_prix)  
(sorte\_de centime unite\_prix)

(sorte\_de unite\_taille unite)  
(sorte\_de metre unite\_taille)  
(sorte\_de centimetre unite\_taille)

(sorte\_de unite\_temps unite)  
(sorte\_de an unite\_temps)  
(sorte\_de mois unite\_temps)  
(sorte\_de jour unite\_temps)  
(sorte\_de heure unite\_temps)

(sorte\_de residence coordonnee)  
(sorte\_de telephone coordonnee)  
(sorte\_de adresse coordonnee)

(sorte\_de nom identite)  
(sorte\_de nom\_jf identite)

(sorte\_de chretien religion)  
(sorte\_de musulman religion)  
(sorte\_de juif religion)  
(sorte\_de athee religion)  
(sorte\_de hindou religion)  
(sorte\_de boudhiste religion)

(sorte\_de couleur etat)  
(sorte\_de bleu couleur)  
(sorte\_de rouge couleur)  
(sorte\_de jaune couleur)  
(sorte\_de rose couleur)  
(sorte\_de gris couleur)  
(sorte\_de noir couleur)  
(sorte\_de blanc couleur)

(sorte\_de qualificatif etat)  
(sorte\_de ancien qualificatif)  
(sorte\_de nouveau qualificatif)  
(sorte\_de grand qualificatif)  
(sorte\_de moyen qualificatif)  
(sorte\_de petit qualificatif)  
(sorte\_de prorogable qualificatif)  
(sorte\_de renouvelable qualificatif)  
(sorte\_de perime qualificatif)  
(sorte\_de valide qualificatif)

(sorte\_de etat\_legal etat)  
(sorte\_de responsabilite etat\_legal)  
(sorte\_de majeur responsabilite)  
(sorte\_de mineur responsabilite)

(sorte\_de nationalite etat)

(sorte\_de situation etat)  
(sorte\_de marie situation)  
(sorte\_de divorce situation)  
(sorte\_de veuf situation)  
(sorte\_de celibataire situation)  
(sorte\_de remarie situation)

(sorte\_de sexe etat)  
(sorte\_de masculin sexe)  
(sorte\_de feminin sexe)

(sorte\_de changement etat)

(agent aller personne)  
(direction aller lieu)

(agent renouveler personne)  
(theme renouveler document)

(agent modifier personne)  
(theme modifier document)

(agent etablir administration)  
(theme etablir document)

(beneficiaire obtenir personne)  
(theme obtenir document)

(agent declarer personne)  
(theme declarer evenement)

(agent remplir personne)  
(theme remplir formulaire)

(agent munir personne)  
(theme munir document)

(agent fournir personne)  
(theme fournir document)  
(beneficiaire fournir administration)

(agent presenter personne)  
(theme presenter piece\_administrative)

(agent accompagner personne)  
(client accompagner personne)

(agent joindre anime)  
(theme joindre anime)

(agent ecouter anime)  
(theme ecouter anime)

(agent donner anime)  
(theme donner inanime)  
(beneficiaire donner anime)

(agent demander anime)  
(theme demander inanime)  
(referent demander anime)

(agent ecrire anime)  
(theme ecrire lettre)  
(beneficiaire ecrire anime)

(agent rencontrer anime)  
(theme rencontrer anime)

(agent appeler anime)  
(theme appeler anime)  
(moyen appeler telephone)

(agent prouver anime)  
(theme prouver document)  
(connaissance prouver etat)

(theme etre\_departement ville)  
(referent etre\_departement departement)

)

## Les déictiques

((jean personne)  
(marie personne)  
(pierre personne)  
(paul personne)  
(document document)  
(aller aller)  
(locuteur locuteur)  
(cri inanime)  
)

## **Annexe E**

### **Résultats obtenus**

## Exemples de créations d'ensembles

(creens 'in 'disjt '(visa lettre document))

Voici la liste des ensembles :

-----  
Ens n 1 : [233 (N = 1) IN NIL (DOCUMENT)]  
Ens n 2 : [231 (N = 1) IN NIL (LETTRE)]  
Ens n 3 : [229 (N = 1) IN NIL (VISA)]  
Ens n 4 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
Ens n 5 : [224 (N = 1) IN NIL (CARTE\_SS)]  
Ens n 6 : [222 (N = 1) IN NIL (VISA)]  
Ens n 7 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
Ens n 8 : [216 (N = 1) CA NIL (NANCY)]  
Ens n 9 : [214 (N = 1) CA NIL (VILLE)]  
Ens n 10 : [210 (N = 2) CA COLLT (VILLE NANCY)]  
Ens n 11 : [194 (N = 1) DE NIL (JEAN)]  
Ens n 12 : [195 (N = 1) DE NIL (MARIE)]  
Ens n 13 : [196 (N = 1) DE NIL (PIERRE)]  
Ens n 14 : [197 (N = 1) DE NIL (PAUL)]  
Ens n 15 : [198 (N = 1) DE NIL (DOCUMENT)]  
Ens n 16 : [199 (N = 1) DE NIL (ALLER)]  
Ens n 17 : [200 (N = 1) DE NIL (LOCUTEUR)]  
Ens n 18 : [201 (N = 1) DE NIL (CRI)]

La référence a été créée

Ref : Ensemble : Concept generique

-----  
234 : [233 (N = 1) IN NIL (DOCUMENT)] : DOCUMENT  
232 : [231 (N = 1) IN NIL (LETTRE)] : LETTRE  
230 : [229 (N = 1) IN NIL (VISA)] : VISA  
228 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)] : DOCUMENT  
225 : [224 (N = 1) IN NIL (CARTE\_SS)] : CARTE\_SS  
223 : [222 (N = 1) IN NIL (VISA)] : VISA  
221 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)] :  
PIECE\_ADMINISTRATIVE  
217 : [216 (N = 1) CA NIL (NANCY)] : NANCY



215 : [214 (N = 1) CA NIL (VILLE)] : VILLE  
 213 : [210 (N = 2) CA COLLT (VILLE NANCY)] : VILLE  
 202 : [194 (N = 1) DE NIL (JEAN)] : PERSONNE  
 203 : [195 (N = 1) DE NIL (MARIE)] : PERSONNE  
 204 : [196 (N = 1) DE NIL (PIERRE)] : PERSONNE  
 205 : [197 (N = 1) DE NIL (PAUL)] : PERSONNE  
 206 : [198 (N = 1) DE NIL (DOCUMENT)] : DOCUMENT  
 207 : [199 (N = 1) DE NIL (ALLER)] : ALLER  
 208 : [200 (N = 1) DE NIL (LOCUTEUR)] : LOCUTEUR  
 209 : [201 (N = 1) DE NIL (CRI)] : INANIME

Voici la liste des relations entre les ensembles :

-----  
 [233 (N = 1) IN NIL (DOCUMENT)] => (INCLUS) => [226 (N = 3) IN  
 DISJT (VISA LETTRE DOCUMENT)]  
 [231 (N = 1) IN NIL (LETTRE)] => (INCLUS) => [226 (N = 3) IN DISJT  
 (VISA LETTRE DOCUMENT)]  
 [229 (N = 1) IN NIL (VISA)] => (INCLUS) => [226 (N = 3) IN DISJT  
 (VISA LETTRE DOCUMENT)]  
 [224 (N = 1) IN NIL (CARTE\_SS)] => (INCLUS) => [218 (N = 2) IN  
 COLLT (VISA CARTE\_SS)]  
 [222 (N = 1) IN NIL (VISA)] => (INCLUS) => [218 (N = 2) IN COLLT  
 (VISA CARTE\_SS)]  
 [216 (N = 1) CA NIL (NANCY)] => (INCLUS) => [210 (N = 2) CA COLLT  
 (VILLE NANCY)]  
 [214 (N = 1) CA NIL (VILLE)] => (INCLUS) => [210 (N = 2) CA COLLT  
 (VILLE NANCY)]  
 -----

(creens 'nd 'collt '(mairie moselle nancy laxou ville))

Voici la liste des ensembles :

-----  
 Ens n 1 : [248 (N = 1) ND NIL (VILLE)]  
 Ens n 2 : [246 (N = 1) ND NIL (LAXOU)]  
 Ens n 3 : [244 (N = 1) ND NIL (NANCY)]  
 Ens n 4 : [242 (N = 1) ND NIL (MOSELLE)]  
 Ens n 5 : [240 (N = 1) ND NIL (MAIRIE)]  
 Ens n 6 : [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU  
 VILLE)]  
 Ens n 7 : [233 (N = 1) IN NIL (DOCUMENT)]

Ens n 8 : [231 (N = 1) IN NIL (LETTRE)]  
 Ens n 9 : [229 (N = 1) IN NIL (VISA)]  
 Ens n 10 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
 Ens n 11 : [224 (N = 1) IN NIL (CARTE\_SS)]  
 Ens n 12 : [222 (N = 1) IN NIL (VISA)]  
 Ens n 13 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
 Ens n 14 : [216 (N = 1) CA NIL (NANCY)]  
 Ens n 15 : [214 (N = 1) CA NIL (VILLE)]  
 Ens n 16 : [210 (N = 2) CA COLLT (VILLE NANCY)]  
 Ens n 17 : [194 (N = 1) DE NIL (JEAN)]  
 Ens n 18 : [195 (N = 1) DE NIL (MARIE)]  
 Ens n 19 : [196 (N = 1) DE NIL (PIERRE)]  
 Ens n 20 : [197 (N = 1) DE NIL (PAUL)]  
 Ens n 21 : [198 (N = 1) DE NIL (DOCUMENT)]  
 Ens n 22 : [199 (N = 1) DE NIL (ALLER)]  
 Ens n 23 : [200 (N = 1) DE NIL (LOCUTEUR)]  
 Ens n 24 : [201 (N = 1) DE NIL (CRI)]

La référence a été créée

Ref : Ensemble : Concept generique

-----  
 249 : [248 (N = 1) ND NIL (VILLE)] : VILLE  
 247 : [246 (N = 1) ND NIL (LAXOU)] : LAXOU  
 245 : [244 (N = 1) ND NIL (NANCY)] : NANCY  
 243 : [242 (N = 1) ND NIL (MOSELLE)] : MOSELLE  
 241 : [240 (N = 1) ND NIL (MAIRIE)] : MAIRIE  
 239 : [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU  
 VILLE)] : LIEU  
 234 : [233 (N = 1) IN NIL (DOCUMENT)] : DOCUMENT  
 232 : [231 (N = 1) IN NIL (LETTRE)] : LETTRE  
 230 : [229 (N = 1) IN NIL (VISA)] : VISA  
 228 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)] : DOCUMENT  
 225 : [224 (N = 1) IN NIL (CARTE\_SS)] : CARTE\_SS  
 223 : [222 (N = 1) IN NIL (VISA)] : VISA  
 221 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)] :  
 PIECE\_ADMINISTRATIVE  
 217 : [216 (N = 1) CA NIL (NANCY)] : NANCY  
 215 : [214 (N = 1) CA NIL (VILLE)] : VILLE  
 213 : [210 (N = 2) CA COLLT (VILLE NANCY)] : VILLE  
 202 : [194 (N = 1) DE NIL (JEAN)] : PERSONNE  
 203 : [195 (N = 1) DE NIL (MARIE)] : PERSONNE  
 204 : [196 (N = 1) DE NIL (PIERRE)] : PERSONNE  
 205 : [197 (N = 1) DE NIL (PAUL)] : PERSONNE

206 : [198 (N = 1) DE NIL (DOCUMENT)] : DOCUMENT  
 207 : [199 (N = 1) DE NIL (ALLER)] : ALLER  
 208 : [200 (N = 1) DE NIL (LOCUTEUR)] : LOCUTEUR  
 209 : [201 (N = 1) DE NIL (CRI)] : INANIME

Voici la liste des relations entre les ensembles :

-----

[248 (N = 1) ND NIL (VILLE)] => (INCLUS) => [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [246 (N = 1) ND NIL (LAXOU)] => (INCLUS) => [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [244 (N = 1) ND NIL (NANCY)] => (INCLUS) => [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [242 (N = 1) ND NIL (MOSELLE)] => (INCLUS) => [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [240 (N = 1) ND NIL (MAIRIE)] => (INCLUS) => [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [233 (N = 1) IN NIL (DOCUMENT)] => (INCLUS) => [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
 [231 (N = 1) IN NIL (LETTRE)] => (INCLUS) => [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
 [229 (N = 1) IN NIL (VISA)] => (INCLUS) => [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
 [224 (N = 1) IN NIL (CARTE\_SS)] => (INCLUS) => [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
 [222 (N = 1) IN NIL (VISA)] => (INCLUS) => [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
 [216 (N = 1) CA NIL (NANCY)] => (INCLUS) => [210 (N = 2) CA COLLT (VILLE NANCY)]  
 [214 (N = 1) CA NIL (VILLE)] => (INCLUS) => [210 (N = 2) CA COLLT (VILLE NANCY)]

-----

(creens 'ca 'disjt '(gendarmerie metz france taxe impots nancy passeport))

Voici la liste des ensembles :

-----

Ens n 1 : [272 (N = 1) CA NIL (PASSEPORT)]  
 Ens n 2 : [270 (N = 1) CA NIL (NANCY)]  
 Ens n 3 : [268 (N = 1) CA NIL (IMPOTS)]  
 Ens n 4 : [266 (N = 1) CA NIL (TAXE)]  
 Ens n 5 : [264 (N = 1) CA NIL (FRANCE)]

Ens n 6 : [262 (N = 1) CA NIL (METZ)]  
 Ens n 7 : [260 (N = 1) CA NIL (GENDARMERIE)]  
 Ens n 8 : [257 (N = 7) CA DISJT (GENDARMERIE METZ FRANCE TAXE  
 IMPOTS NANCY PASSEPORT)]  
 Ens n 9 : [248 (N = 1) ND NIL (VILLE)]  
 Ens n 10 : [246 (N = 1) ND NIL (LAXOU)]  
 Ens n 11 : [244 (N = 1) ND NIL (NANCY)]  
 Ens n 12 : [242 (N = 1) ND NIL (MOSELLE)]  
 Ens n 13 : [240 (N = 1) ND NIL (MAIRIE)]  
 Ens n 14 : [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU  
 VILLE)]  
 Ens n 15 : [233 (N = 1) IN NIL (DOCUMENT)]  
 Ens n 16 : [231 (N = 1) IN NIL (LETTRE)]  
 Ens n 17 : [229 (N = 1) IN NIL (VISA)]  
 Ens n 18 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)]  
 Ens n 19 : [224 (N = 1) IN NIL (CARTE\_SS)]  
 Ens n 20 : [222 (N = 1) IN NIL (VISA)]  
 Ens n 21 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)]  
 Ens n 22 : [216 (N = 1) CA NIL (NANCY)]  
 Ens n 23 : [214 (N = 1) CA NIL (VILLE)]  
 Ens n 24 : [210 (N = 2) CA COLLT (VILLE NANCY)]  
 Ens n 25 : [194 (N = 1) DE NIL (JEAN)]  
 Ens n 26 : [195 (N = 1) DE NIL (MARIE)]  
 Ens n 27 : [196 (N = 1) DE NIL (PIERRE)]  
 Ens n 28 : [197 (N = 1) DE NIL (PAUL)]  
 Ens n 29 : [198 (N = 1) DE NIL (DOCUMENT)]  
 Ens n 30 : [199 (N = 1) DE NIL (ALLER)]  
 Ens n 31 : [200 (N = 1) DE NIL (LOCUTEUR)]  
 Ens n 32 : [201 (N = 1) DE NIL (CRI)]

La référence a été créée

Ref : Ensemble : Concept generique

-----  
 273 : [272 (N = 1) CA NIL (PASSEPORT)] : PASSEPORT  
 271 : [270 (N = 1) CA NIL (NANCY)] : NANCY  
 269 : [268 (N = 1) CA NIL (IMPOTS)] : IMPOTS  
 267 : [266 (N = 1) CA NIL (TAXE)] : TAXE  
 265 : [264 (N = 1) CA NIL (FRANCE)] : FRANCE  
 263 : [262 (N = 1) CA NIL (METZ)] : METZ  
 261 : [260 (N = 1) CA NIL (GENDARMERIE)] : GENDARMERIE  
 259 : [257 (N = 7) CA DISJT (GENDARMERIE METZ FRANCE TAXE  
 IMPOTS NANCY PASSEPORT)] : INANIME  
 249 : [248 (N = 1) ND NIL (VILLE)] : VILLE

247 : [246 (N = 1) ND NIL (LAXOU)] : LAXOU  
 245 : [244 (N = 1) ND NIL (NANCY)] : NANCY  
 243 : [242 (N = 1) ND NIL (MOSELLE)] : MOSELLE  
 241 : [240 (N = 1) ND NIL (MAIRIE)] : MAIRIE  
 239 : [235 (N = 5) ND COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)] : LIEU  
 234 : [233 (N = 1) IN NIL (DOCUMENT)] : DOCUMENT  
 232 : [231 (N = 1) IN NIL (LETTRE)] : LETTRE  
 230 : [229 (N = 1) IN NIL (VISA)] : VISA  
 228 : [226 (N = 3) IN DISJT (VISA LETTRE DOCUMENT)] : DOCUMENT  
 225 : [224 (N = 1) IN NIL (CARTE\_SS)] : CARTE\_SS  
 223 : [222 (N = 1) IN NIL (VISA)] : VISA  
 221 : [218 (N = 2) IN COLLT (VISA CARTE\_SS)] :  
 PIECE\_ADMINISTRATIVE  
 217 : [216 (N = 1) CA NIL (NANCY)] : NANCY  
 215 : [214 (N = 1) CA NIL (VILLE)] : VILLE  
 213 : [210 (N = 2) CA COLLT (VILLE NANCY)] : VILLE  
 202 : [194 (N = 1) DE NIL (JEAN)] : PERSONNE  
 203 : [195 (N = 1) DE NIL (MARIE)] : PERSONNE  
 204 : [196 (N = 1) DE NIL (PIERRE)] : PERSONNE  
 205 : [197 (N = 1) DE NIL (PAUL)] : PERSONNE  
 206 : [198 (N = 1) DE NIL (DOCUMENT)] : DOCUMENT  
 207 : [199 (N = 1) DE NIL (ALLER)] : ALLER  
 208 : [200 (N = 1) DE NIL (LOCUTEUR)] : LOCUTEUR  
 209 : [201 (N = 1) DE NIL (CRI)] : INANIME

Voici la liste des relations entre les ensembles :

-----  
 [272 (N = 1) CA NIL (PASSEPORT)] => (INCLUS) => [257 (N = 7) CA  
 DISJT (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY  
 PASSEPORT)]  
 [270 (N = 1) CA NIL (NANCY)] => (INCLUS) => [257 (N = 7) CA DISJT  
 (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY PASSEPORT)]  
 [268 (N = 1) CA NIL (IMPOTS)] => (INCLUS) => [257 (N = 7) CA DISJT  
 (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY PASSEPORT)]  
 [266 (N = 1) CA NIL (TAXE)] => (INCLUS) => [257 (N = 7) CA DISJT  
 (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY PASSEPORT)]  
 [264 (N = 1) CA NIL (FRANCE)] => (INCLUS) => [257 (N = 7) CA DISJT  
 (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY PASSEPORT)]  
 [262 (N = 1) CA NIL (METZ)] => (INCLUS) => [257 (N = 7) CA DISJT  
 (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY PASSEPORT)]

[260 (N = 1) CA NIL (GENDARMERIE)] => (INCLUS) => [257 (N = 7) CA  
 DISJT (GENDARMERIE METZ FRANCE TAXE IMPOTS NANCY  
 PASSEPORT)]  
 [248 (N = 1) ND NIL (VILLE)] => (INCLUS) => [235 (N = 5) ND COLLT  
 (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [246 (N = 1) ND NIL (LAXOU)] => (INCLUS) => [235 (N = 5) ND COLLT  
 (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [244 (N = 1) ND NIL (NANCY)] => (INCLUS) => [235 (N = 5) ND COLLT  
 (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [242 (N = 1) ND NIL (MOSELLE)] => (INCLUS) => [235 (N = 5) ND  
 COLLT (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [240 (N = 1) ND NIL (MAIRIE)] => (INCLUS) => [235 (N = 5) ND COLLT  
 (MAIRIE MOSELLE NANCY LAXOU VILLE)]  
 [233 (N = 1) IN NIL (DOCUMENT)] => (INCLUS) => [226 (N = 3) IN  
 DISJT (VISA LETTRE DOCUMENT)]  
 [231 (N = 1) IN NIL (LETTRE)] => (INCLUS) => [226 (N = 3) IN DISJT  
 (VISA LETTRE DOCUMENT)]  
 [229 (N = 1) IN NIL (VISA)] => (INCLUS) => [226 (N = 3) IN DISJT  
 (VISA LETTRE DOCUMENT)]  
 [224 (N = 1) IN NIL (CARTE\_SS)] => (INCLUS) => [218 (N = 2) IN  
 COLLT (VISA CARTE\_SS)]  
 [222 (N = 1) IN NIL (VISA)] => (INCLUS) => [218 (N = 2) IN COLLT  
 (VISA CARTE\_SS)]  
 [216 (N = 1) CA NIL (NANCY)] => (INCLUS) => [210 (N = 2) CA COLLT  
 (VILLE NANCY)]  
 [214 (N = 1) CA NIL (VILLE)] => (INCLUS) => [210 (N = 2) CA COLLT  
 (VILLE NANCY)]

## Les primitives supérieures

(164 MVMET\_2)  
(165 ACT\_2)  
(166 ACT\_2)  
(167 ACT\_2)  
(168 COGNT\_3)  
(169 ACT\_2)  
(170 ECHOBT\_2)  
(171 COGNT\_3)  
(172 ACT\_2)  
(173 ACT\_2)  
(174 ECHPROD\_3)  
(175 ACT\_2)  
(176 ACT\_2)  
(177 ACT\_2)  
(178 COGNT\_1)  
(178 COGNT\_2)  
(179 ECHPROD\_3)  
(180 COGNT\_3)  
(181 ACT\_2)  
(183 COGNT\_3)  
(184 COGNT\_3)  
(185 STATUT\_2)

## Les concepts génériques

(1 INANIME)  
(2 LIEU)  
(3 VILLE)  
(4 NANCY)  
(5 VANDOEUVRE)  
(6 VILLERS)  
(7 LAXOU)  
(8 MAXEVILLE)  
(9 MALZEVILLE)  
(10 PARIS)  
(11 METZ)  
(12 DEPARTEMENT)  
(13 MOSELLE)  
(14 MEURTHE\_ET\_MOSELLE)  
(15 MEUSE)

(16 VOSGES)  
(17 REGION\_PARISIENNE)  
(18 COMMERCE)  
(19 BUREAU\_DE\_TABAC)  
(20 SUPERMARCHE)  
(21 ADMINISTRATION)  
(22 SP\_P)  
(23 PREFECTURE)  
(24 SOUS\_PREFECTURE)  
(25 MAIRIE)  
(26 HOTEL\_DE\_VILLE)  
(27 COMMISSARIAT)  
(28 GENDARMERIE)  
(29 CIRA)  
(30 CPAM)  
(31 ASINA)  
(32 ASS\_CONSOMMATEURS)  
(33 OPHLM)  
(34 CONSULAT)  
(35 CONSULAT\_D\_ALGERIE)  
(36 CONSULAT\_DE\_RFA)  
(37 CONSULAT\_DU\_MAROC)  
(38 PAYS)  
(39 FRANCE)  
(40 ETRANGER)  
(41 CEE)  
(42 BELGIQUE)  
(43 LUXEMBOURG)  
(44 RFA)  
(45 ITALIE)  
(46 PAYS\_BAS)  
(47 ESPAGNE)  
(48 PORTUGAL)  
(49 GRECE)  
(50 DANEMARK)  
(51 GRANDE\_BRETAGNE)  
(52 ANGLETERRE)  
(53 ECOSSE)  
(54 IRLANDE)  
(55 PAYS\_DE\_GALLES)  
(56 NON\_CEE)  
(57 USA)  
(58 URSS)  
(59 MADAGASCAR)  
(60 MAROC)  
(61 ALGERIE)



(62 DOCUMENT)  
(63 PIECE\_ADMINISTRATIVE)  
(64 CARTE\_D\_IDENTITE)  
(66 PASSEPORT)  
(67 CARTE\_GRISE)  
(68 FICHE\_D\_ETAT\_CIVIL))  
(70 LIVRET\_DE\_FAMILLE)  
(71 CARTE\_DE\_SEJOUR)  
(72 CERTIFICAT\_DE\_SCOLARITE)  
(73 AUTORISATION\_DE\_SORTIE)  
(74 DECLARATION\_DE\_VOL)  
(75 DECLARATION\_DE\_PERTE)  
(76 VISA)  
(77 CARTE\_SS)  
(78 CERTIFICAT\_MEDICAL)  
(79 LETTRE)  
(80 FORMULAIRE)  
(81 PHOTO)  
(82 TIMBRE)  
(83 TAXE)  
(84 IMPOTS)  
(85 TVA)  
(86 IMPOTS\_LOCAUX)  
(87 EVENEMENT)  
(88 DECES)  
(89 MARIAGE)  
(90 NAISSANCE)  
(91 PERTE)  
(92 VOL)  
(93 ANIME)  
(94 PERSONNE)  
(95 LOCUTEUR)  
(96 MACHINE)  
(97 TIERS)  
(98 ANIMAL)  
(99 DOMESTIQUE)  
(100 CHIEN)  
(101 CHAT)  
(102 POISSON)  
(103 SAUVAGE)  
(104 UNITE)  
(105 UNITE\_PRIX)  
(106 FRANC)  
(107 CENTIME)  
(108 UNITE\_TAILLE)  
(109 METRE)

(110 CENTIMETRE)  
(111 UNITE\_TEMPS)  
(112 AN)  
(113 MOIS)  
(114 JOUR)  
(115 HEURE)  
(116 COORDONNEE)  
(117 RESIDENCE)  
(118 TELEPHONE)  
(119 ADRESSE)  
(120 IDENTITE)  
(121 NOM)  
(122 NOM\_JF)  
(123 RELIGION)  
(124 CHRETIEN)  
(125 MUSULMAN)  
(126 JUIF)  
(127 ATHEE)  
(128 HINDOU)  
(129 BOUDHISTE)  
(130 ETAT)  
(131 COULEUR)  
(132 BLEU)  
(133 ROUGE)  
(134 JAUNE)  
(135 ROSE)  
(136 GRIS)  
(137 NOIR)  
(138 BLANC)  
(139 QUALIFICATIF)  
(140 ANCIEN)  
(141 NOUVEAU)  
(142 GRAND)  
(143 MOYEN)  
(144 PETIT)  
(145 PROROGABLE)  
(146 RENOUVELABLE)  
(147 PERIME)  
(148 VALIDE)  
(149 ETAT\_LEGAL)  
(150 RESPONSABILITE)  
(151 MAJEUR)  
(152 MINEUR)  
(153 NATIONALITE)  
(154 SITUATION)  
(155 MARIE)

(156 DIVORCE)  
(157 VEUF)  
(158 CELIBATAIRE)  
(159 REMARIE)  
(160 SEXE)  
(161 MASCULIN)  
(162 FEMININ)  
(163 CHANGEMENT)  
(164 ALLER)  
(165 RENOUVELER)  
(166 MODIFIER)  
(167 ETABLIR)  
(168 AUTORISER)  
(169 ANNULER)  
(170 OBTENIR)  
(171 DECLARER)  
(172 REMPLIR)  
(173 MUNIR)  
(174 FOURNIR)  
(175 PRESENTER)  
(176 ACCOMPAGNER)  
(177 JOINDRE)  
(178 ECOUTER)  
(179 DONNER)  
(180 DEMANDER)  
(181 ECRIRE)  
(182 RENCONTRER)  
(183 APPELER)  
(184 PROUVER)  
(185 ETRE\_DEPARTEMENT)

### **Les relations entre les concepts génériques**

(SORTE\_DE 2 1)  
(SORTE\_DE 3 2)  
(SORTE\_DE 4 3)  
(SORTE\_DE 5 3)  
(SORTE\_DE 6 3)  
(SORTE\_DE 7 3)  
(SORTE\_DE 9 3)  
(SORTE\_DE 10 3)  
(SORTE\_DE 11 3)  
(SORTE\_DE 12 2)  
(SORTE\_DE 13 12)  
(SORTE\_DE 14 12)

(SORTE\_DE 15 12)  
(SORTE\_DE 16 12)  
(SORTE\_DE 17 12)  
(SORTE\_DE 18 2)  
(SORTE\_DE 19 18)  
(SORTE\_DE 20 18)  
(SORTE\_DE 21 2)  
(SORTE\_DE 22 21)  
(SORTE\_DE 23 22)  
(SORTE\_DE 24 22)  
(SORTE\_DE 25 21)  
(SORTE\_DE 26 21)  
(SORTE\_DE 27 21)  
(SORTE\_DE 28 21)  
(SORTE\_DE 29 21)  
(SORTE\_DE 30 21)  
(SORTE\_DE 31 21)  
(SORTE\_DE 32 21)  
(SORTE\_DE 33 21)  
(SORTE\_DE 34 21)  
(SORTE\_DE 35 34)  
(SORTE\_DE 36 34)  
(SORTE\_DE 37 34)  
(SORTE\_DE 38 2)  
(SORTE\_DE 39 38)  
(SORTE\_DE 40 38)  
(SORTE\_DE 41 40)  
(SORTE\_DE 42 41)  
(SORTE\_DE 43 41)  
(SORTE\_DE 44 41)  
(SORTE\_DE 45 41)  
(SORTE\_DE 46 41)  
(SORTE\_DE 47 41)  
(SORTE\_DE 48 41)  
(SORTE\_DE 49 41)  
(SORTE\_DE 50 41)  
(SORTE\_DE 51 41)  
(SORTE\_DE 52 51)  
(SORTE\_DE 53 51)  
(SORTE\_DE 54 51)  
(SORTE\_DE 55 51)  
(SORTE\_DE 56 40)  
(SORTE\_DE 57 56)  
(SORTE\_DE 58 56)  
(SORTE\_DE 59 56)  
(SORTE\_DE 60 56)

(SORTE\_DE 61 56)  
(SORTE\_DE 62 1)  
(SORTE\_DE 63 62)  
(SORTE\_DE 66 63)  
(SORTE\_DE 67 63)  
(SORTE\_DE 70 63)  
(SORTE\_DE 71 63)  
(SORTE\_DE 72 63)  
(SORTE\_DE 73 63)  
(SORTE\_DE 74 63)  
(SORTE\_DE 75 63)  
(SORTE\_DE 76 63)  
(SORTE\_DE 77 63)  
(SORTE\_DE 78 62)  
(SORTE\_DE 79 62)  
(SORTE\_DE 81 62)  
(SORTE\_DE 82 62)  
(SORTE\_DE 83 1)  
(SORTE\_DE 84 83)  
(SORTE\_DE 85 83)  
(SORTE\_DE 86 83)  
(SORTE\_DE 88 87)  
(SORTE\_DE 89 87)  
(SORTE\_DE 90 87)  
(SORTE\_DE 91 87)  
(SORTE\_DE 92 87)  
(SORTE\_DE 94 93)  
(SORTE\_DE 95 94)  
(SORTE\_DE 96 94)  
(SORTE\_DE 97 94)  
(SORTE\_DE 98 93)  
(SORTE\_DE 99 98)  
(SORTE\_DE 100 99)  
(SORTE\_DE 101 99)  
(SORTE\_DE 102 99)  
(SORTE\_DE 103 98)  
(SORTE\_DE 21 93)  
(SORTE\_DE 105 104)  
(SORTE\_DE 106 105)  
(SORTE\_DE 107 105)  
(SORTE\_DE 108 104)  
(SORTE\_DE 109 108)  
(SORTE\_DE 110 108)  
(SORTE\_DE 111 104)  
(SORTE\_DE 112 111)  
(SORTE\_DE 113 111)

(SORTE\_DE 114 111)  
(SORTE\_DE 115 111)  
(SORTE\_DE 117 116)  
(SORTE\_DE 118 116)  
(SORTE\_DE 119 116)  
(SORTE\_DE 121 120)  
(SORTE\_DE 122 120)  
(SORTE\_DE 124 123)  
(SORTE\_DE 125 123)  
(SORTE\_DE 126 123)  
(SORTE\_DE 127 123)  
(SORTE\_DE 128 123)  
(SORTE\_DE 129 123)  
(SORTE\_DE 131 130)  
(SORTE\_DE 132 131)  
(SORTE\_DE 133 131)  
(SORTE\_DE 134 131)  
(SORTE\_DE 135 131)  
(SORTE\_DE 136 131)  
(SORTE\_DE 137 131)  
(SORTE\_DE 138 131)  
(SORTE\_DE 139 130)  
(SORTE\_DE 140 139)  
(SORTE\_DE 141 139)  
(SORTE\_DE 142 139)  
(SORTE\_DE 143 139)  
(SORTE\_DE 144 139)  
(SORTE\_DE 145 139)  
(SORTE\_DE 146 139)  
(SORTE\_DE 147 139)  
(SORTE\_DE 148 139)  
(SORTE\_DE 149 130)  
(SORTE\_DE 150 149)  
(SORTE\_DE 151 150)  
(SORTE\_DE 152 150)  
(SORTE\_DE 153 130)  
(SORTE\_DE 154 130)  
(SORTE\_DE 155 154)  
(SORTE\_DE 156 154)  
(SORTE\_DE 157 154)  
(SORTE\_DE 158 154)  
(SORTE\_DE 159 154)  
(SORTE\_DE 160 130)  
(SORTE\_DE 161 160)  
(SORTE\_DE 162 160)  
(SORTE\_DE 163 130)

(AGENT 164 94)  
(DIRECTION 164 2)  
(AGENT 165 94)  
(THEME 165 62)  
(AGENT 166 94)  
(THEME 166 62)  
(AGENT 167 21)  
(THEME 167 62)  
(BENEFICIAIRE 170 94)  
(THEME 170 62)  
(AGENT 171 94)  
(THEME 171 87)  
(AGENT 172 94)  
(THEME 172 80)  
(AGENT 173 94)  
(THEME 173 62)  
(AGENT 174 94)  
(THEME 174 62)  
(BENEFICIAIRE 174 21)  
(AGENT 175 94)  
(THEME 175 63)  
(AGENT 176 94)  
(CLIENT 176 94)  
(AGENT 177 93)  
(THEME 177 93)  
(AGENT 178 93)  
(THEME 178 93)  
(AGENT 179 93)  
(THEME 179 1)  
(BENEFICIAIRE 179 93)  
(AGENT 180 93)  
(THEME 180 1)  
(REFERENT 180 93)  
(AGENT 181 93)  
(THEME 181 79)  
(BENEFICIAIRE 181 93)  
(AGENT 182 93)  
(THEME 182 93)  
(AGENT 183 93)  
(THEME 183 93)  
(MOYEN 183 118)  
(AGENT 184 93)  
(THEME 184 62)  
(CONNAISSANCE 184 130)  
(THEME 185 3)  
(REFERENT 185 12)

## **Les concepts individuels déictiques**

(186 JEAN)  
(187 MARIE)  
(188 PIERRE)  
(189 PAUL)  
(190 DOCUMENT)  
(191 ALLER)  
(192 LOCUTEUR)  
(193 CRI)

## **Les ensembles déictiques**

(194 (N = 1) DE NIL (186))  
(195 (N = 1) DE NIL (187))  
(196 (N = 1) DE NIL (188))  
(197 (N = 1) DE NIL (189))  
(198 (N = 1) DE NIL (190))  
(199 (N = 1) DE NIL (191))  
(200 (N = 1) DE NIL (192))  
(201 (N = 1) DE NIL (193))

## **Les références entre les déictiques et leur concept générique**

(202 194 94)  
(203 195 94)  
(204 196 94)  
(205 197 94)  
(206 198 62)  
(207 199 164)  
(208 200 95)  
(209 201 1)



## Les primitives de construction d'une formule

